

# **Modellierung modularer Materialfluss-Systeme mit Hilfe von künstlichen neuronalen Netzen**

Von der Fakultät Maschinenwesen  
der  
Technischen Universität Dresden  
zur

Erlangung des akademischen Grades  
Doktoringenieur (Dr.-Ing.)  
angenommene Dissertation

Dipl.-Math. Ulf Markwardt  
geboren am 27.11.1969 in Berlin

Tag der Einreichung: 4.11.2003

Tag der Verteidigung: 29.9.2004

Gutachter: Prof. Dr.-Ing.habil. Hans-Georg Marquardt  
Prof. Dr.-Ing.habil. Kai Furmans  
Prof. Dr.-Ing. Rainer Bruns

Prof. Dr.-Ing. habil. Volker Ulbricht  
Vorsitzender der Promotionskommission



# Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Fördertechnik, Baumaschinen und Logistik der TU Dresden. Das ihr zugrunde liegende Forschungsvorhaben „Neuronale Netze zur Verhaltensprognose von Materialfluss-Systemen“ wurde aus Haushaltsmitteln des Bundesministeriums für Wirtschaft und Technologie (BMWi) über die Arbeitsgemeinschaft industrieller Forschungsvereinigungen „Otto von Guericke“ e.V. (AiF) gefördert (AiF-FV-Nr. 13009 BR) und im Auftrag der Bundesvereinigung Logistik e.V. (BVL) ausgeführt.

Ich möchte mich bei Prof. Dr.-Ing. habil. Hans-Georg Marquardt für die Betreuung dieser Arbeit und wichtige Impulse dafür bedanken.

Stellvertretend für meine Kollegen bedanke ich mich bei Dipl.-Wirtsch.-Ing. Michael Glass, Dr.-Ing. Frank Schulze und Dipl.-Ing. Karsten Turek für viele anregende Diskussionen, bei Dr.-Ing. Ingolf Meinhardt für schnelle Antworten auf Java-Probleme und bei Dr.-Ing. Thomas Leonhardt für Rat bei den vielen kleinen allgemeinen ingenieurtechnischen Fragen. Dank gilt außerdem Dipl.-Ing. Tobias Nähring für kompetente Beratung bei L<sup>A</sup>T<sub>E</sub>X-Fragen.

Ulf Markwardt

Dresden, den 4. November 2003

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Ausgangssituation . . . . .	1
1.2	Zielstellung . . . . .	2
1.3	Aufbau der Arbeit . . . . .	5
<b>2</b>	<b>Modellierung</b>	<b>6</b>
2.1	Grundlagen . . . . .	6
2.1.1	Systeme . . . . .	6
2.1.2	Modelle . . . . .	8
2.2	Systematik . . . . .	10
2.2.1	Modellzweck . . . . .	10
2.2.2	Art der Modellierung . . . . .	11
2.2.3	Gegenstand der Modellierung . . . . .	12
2.2.4	Methoden . . . . .	12
2.2.5	Gültigkeit eines Modells . . . . .	13
2.3	Modellierung von Materialfluss-Systemen . . . . .	14
2.3.1	Quantile . . . . .	14
2.3.2	Statische Berechnungsmodelle . . . . .	15
2.3.3	Wartesysteme . . . . .	17
2.3.4	Diskrete Simulation . . . . .	18
2.3.5	Vergleich von Modellierungsarten . . . . .	19
2.3.6	Metamodelle . . . . .	19
<b>3</b>	<b>Neuronale Netze</b>	<b>21</b>
3.1	Aufbau neuronaler Netze . . . . .	22
3.2	Training mit Backpropagation . . . . .	24
3.2.1	Einzelner Trainingsschritt . . . . .	25
3.2.2	Trainingszyklen . . . . .	27
3.3	Einsatz neuronaler Netze . . . . .	28
<b>4</b>	<b>Modell für Elemente von Materialfluss-Systemen</b>	<b>31</b>
4.1	Modellbildung . . . . .	32
4.1.1	Lokalitätsbedingungen . . . . .	33
4.1.2	Egalisierung . . . . .	34
4.1.3	Statistische Modellierung der Ströme . . . . .	35
4.1.4	Parametrisches Modell . . . . .	37

4.1.5	Numerische Approximation mit neuronalen Netzen . . . . .	40
4.1.6	Gültigkeit und Korrektheit . . . . .	45
4.2	Modellklassen . . . . .	50
4.2.1	Einschleusung . . . . .	51
4.2.2	Ausschleusung . . . . .	59
4.2.3	Kreuzung . . . . .	61
4.2.4	Regalbediengerät . . . . .	64
4.2.5	Bedienstation . . . . .	68
4.3	Zusammenfassung . . . . .	70
<b>5</b>	<b>Hybride Modellierung</b>	<b>71</b>
5.1	Aufbau eines hybriden Modells . . . . .	71
5.1.1	Berechnung der Parameter . . . . .	73
5.1.2	Zyklische Berechnungsgraphen . . . . .	74
5.2	Beispiele . . . . .	77
5.2.1	Hochregallager . . . . .	77
5.2.2	Kommissionierbereich . . . . .	80
5.2.3	Gesamtsystem mit Lager und Kommissionierbereich . . . . .	83
5.3	Zusammenfassung . . . . .	86
<b>6</b>	<b>Fazit</b>	<b>87</b>
6.1	Ergebnisse der Arbeit . . . . .	87
6.2	Ausblick . . . . .	87
6.2.1	Modellierung weiterer Klassen . . . . .	87
6.2.2	Modellparameter . . . . .	88
6.2.3	Weitere Ideen . . . . .	89
<b>A</b>	<b>Benutzte Werkzeuge</b>	<b>90</b>
A.1	Erzeugung der Trainingsdaten . . . . .	90
A.1.1	Statistik-Bibliothek . . . . .	91
A.2	Entwurf und Training der neuronalen Netze . . . . .	91
A.2.1	Topologie der Netze . . . . .	91
A.2.2	Training der Netze . . . . .	92
A.2.3	Aufbereitung des neuronalen Netzes . . . . .	92
A.3	Der Modellator „TransHyb“ . . . . .	93
A.4	Komplexe diskrete Simulationen . . . . .	94
A.5	Mathematische Auswertung . . . . .	95
<b>B</b>	<b>Fehler der Modelle einzelner Elemente</b>	<b>96</b>
B.1	Ausschleusung . . . . .	97
B.1.1	Variationskoeffizient des Ausgangs der Hauptstrecke . . . . .	97
B.1.2	Variationskoeffizient des Ausgangs der Nebenstrecke . . . . .	97
B.1.3	90%-Quantil der Warteschlange . . . . .	97
B.2	Kreuzung mit FCFS-Steuerung . . . . .	98
B.2.1	Variationskoeffizient des Ausgangs 3 (Hauptstrecke) . . . . .	98

---

B.2.2	Variationskoeffizient des Ausgangs 4 (Nebenstrecke) . . . . .	98
B.2.3	90%-Quantil der Warteschlange (Hauptstrecke) . . . . .	98
B.2.4	90%-Quantil der Warteschlange (Nebenstrecke) . . . . .	99
B.2.5	Auslastung . . . . .	99
B.3	Kreuzung mit Vorfahrtsregelung . . . . .	99
B.3.1	Variationskoeffizient des Ausgangs 3 (Hauptstrecke) . . . . .	99
B.3.2	Variationskoeffizient des Ausgangs 4 (Nebenstrecke) . . . . .	100
B.3.3	90%-Quantil der Warteschlange (Hauptstrecke) . . . . .	100
B.3.4	90%-Quantil der Warteschlange (Nebenstrecke) . . . . .	100
B.3.5	Auslastung . . . . .	101
B.4	Regalbediengerät . . . . .	101
B.4.1	Variationskoeffizient des Ausgangs . . . . .	101
B.4.2	90%-Quantil der Warteschlange vor dem Lager . . . . .	101
B.4.3	Auslastung des Regalbediengeräts . . . . .	102
B.5	Bedienstation . . . . .	102
B.5.1	Variationskoeffizient des Ausgangs . . . . .	102
B.5.2	90%-Quantil der Warteschlange . . . . .	102
<b>C</b>	<b>Verzeichnis verwendeter Symbole</b>	<b>103</b>
	<b>Literaturverzeichnis</b>	<b>106</b>
	<b>Index</b>	<b>112</b>

# 1 Einleitung

## 1.1 Ausgangssituation

Eine hohe Auslastung der jeweils teuersten Ressourcen ist oft eine wichtige Voraussetzung für eine hohe Effizienz. Das gilt sowohl für die Produktion als auch für die Distribution, unabhängig davon, ob es sich dabei um menschliche Arbeitskraft oder eine Fertigungsmaschine handelt.

Hohe Auslastung bedeutet, es müssen ständig Aufträge für die Ressource vorhanden sein. *Puffer* (auch als *Warteräume* oder *Stauplätze* bezeichnet) in einem Materialfluss-System dienen dazu, Fertigungsabschnitte zu entkoppeln, und somit die Wahrscheinlichkeit des Leerlaufs nachgelagerter Anlagen zu verringern. Bei hohem Verkehrsaufkommen sind Stauplätze auch vor Transportsystem-Elementen nötig, um Blockierungen zu vermeiden. In dieser Arbeit wird ein neues Modell vorgestellt, das die Dimensionierung von Stauplätzen in Materialfluss-Systemen unterstützt.

Für eine bessere Verständlichkeit werden an dieser Stelle wichtige auftretende Begriffe erklärt.

Die VDI-Richtlinie 2411 [VDI70] definierte im Jahre 1970: „*Materialfluss* ist die Verkettung aller Vorgänge beim Gewinnen, Be- und Verarbeiten, sowie bei der Verteilung von Gütern innerhalb bestimmter Bereiche. Als Materialfluss-System wird ein zwischen Eingang (*input*) und Ausgang (*output*) begrenzter Bereich verstanden.“

Konkreter beschreibt Großeschallau [Gro84] ein *Materialfluss-System* (*Transportsystem*) als ein logistisches Teilsystem, das den Transport von Materialien innerhalb eines technischen Systems realisiert.

Ein *Transportsystem-Element* (auch *Transportsystem-Komponente* oder *Elementarstation* [Gud99]) ist die kleinste immobile Einheit eines Materialfluss-Systems. Beispiele für Transportsystem-Elemente sind Zusammenführungen, Verzweigungen, Regalbediengeräte.

Diese Arbeit beschäftigt sich ausschließlich mit dem Stückgut-Transport innerhalb einer Fabrik oder eines Distributionszentrums: die bewegten Materialien werden als *Transporteinheiten* bzw. *Güter* bezeichnet. Obwohl auch Fahrzeugsysteme bis zu einem gewissen Grade mit den hier vorgestellten Methoden modelliert werden können, werden sie aus Gründen des Umfangs dieser Arbeit nicht diskutiert. Es wird im Folgenden vorausgesetzt, dass der Transport im Wesentlichen mit Stetigförderern erfolgt.

Ein *Gutstrom* ist ein diskreter stochastischer Prozess, dessen zeitliche Ereignisse durch Eintreffen (Eingangsstrom) bzw. Absenden (Abgangsstrom) von Transportein-

heiten definiert sind.

Warteschlangen entwickeln sich, wenn über eine gewisse Zeit an einem Element des Materialfluss-Systems mehr Transporteinheiten eintreffen, als es verarbeiten kann. Gilt das während der gesamten Betriebszeit, liegt eventuell ein systematisches Problem vor, dieser Fall wird hier nicht näher betrachtet.

Hier interessieren vielmehr Warteschlangen, die dadurch entstehen, dass Ankunfts- und Bedienprozesse eines Transportsystem-Elements nicht synchronisiert sind. Ankunftsprozesse an einem Element werden als zufällig betrachtet. Für die Bemessung von Warteräumen gibt es dann im Wesentlichen zwei Möglichkeiten:

### - Theorie der Wartesysteme

Diese Theorie bezieht sich auf die stochastischen Ankunftsprozesse von Transporteinheiten und die Bedienprozesse von Maschinen. Kernstück der Modellierung eines Warteraumes ist dabei die Berechnung der Wahrscheinlichkeiten dafür, dass eine neue Ankunft erfolgt, während eine Forderung bearbeitet wird. Diese Berechnung ist für konkrete Transportsystem-Elemente nicht immer praktikabel, da lokale Steuerungsregeln oft eine analytische Behandlung mit Hilfe von Zustandsgraphen über Markov-Prozesse *de facto* unmöglich machen.

Für die Berechnung von Kenngrößen, wie der mittleren Warteschlangenlänge oder der mittleren Wartezeit, benutzt man jedoch ungeachtet dessen Ergebnisse der Warteschlangentheorie für Markov-Prozesse, um sich eine ungefähre Vorstellung über die Wartesysteme zu verschaffen. Damit besitzt man sehr einfache Berechnungsvorschriften, die jederzeit ohne Hilfsmittel anwendbar sind. Die mittlere Warteschlangenlänge ist jedoch kaum geeignet, um den Stauraum sicher und ökonomisch zu dimensionieren.

Weitere Ausführungen dazu finden sich in Abschnitt 2.3.3.

### - Diskrete Simulation

Ein Werkzeug, mit dem bei Kenntnis der beteiligten Einflussgrößen und Prozesse eine fast beliebig hohe Planungssicherheit gewonnen werden kann, ist die diskrete Simulation. Die Entwicklung eines Modells ist jedoch unter Umständen aufwändig, das betrifft auch Änderungen im Layout der Anlage. Aus diesem Grunde wird die Simulation meist erst für die endgültige Variante zum Leistungsnachweis des Materialfluss-Systems oder zur Optimierung der Steuerung eingesetzt.

In Abschnitt 2.3.4 wird die diskrete Simulation genauer vorgestellt.

## 1.2 Zielstellung

Es ist gegenwärtig keine Methode bekannt, die mit hoher Planungssicherheit eine schnelle Grobplanung von Materialfluss-Systemen mit Hinblick auf die benötigten Stauplätze gestattet. Aus diesem Grunde wird hier ein neues Modell vorgestellt, das es ermöglicht, diese Lücke schließen.



Es geht dabei um die Entwicklung eines einfach handzuhabenden Werkzeuges für schnelle Prognosen über den Bedarf an Stauplätzen und die Auslastung von Ressourcen. Mit dessen Hilfe kann sowohl für eine veränderte Belastungssituation eines bestehenden Systems, wie auch für den Neubau eines Materialfluss-Systems eine verbesserte Planungssicherheit gewährleistet werden.

## Grundidee

Transportsysteme sind modular aufgebaut, das heißt, es gibt dort stets ähnliche, wiederkehrende Elemente. Deren jeweilige Abgangsströme hängen von den zugehörigen Eingangsströmen und lokalen technischen Größen der Elemente ab. Das Gleiche gilt für die Auslastung und Warteschlangen an den Eingängen.

Durch die Wahl einer geeigneten Parameterdarstellung für die Ströme ist eine numerische Berechnung möglich. Das heißt, für gegebene technische Parameter und eine konkrete logistische Belastung eines Elements können die Abgangsströme und interessierende Kenngrößen mit einem numerischen Modell approximiert werden.

Die strukturtreue Verbindung solcher einzelnen Elemente ermöglicht schließlich Aussagen über komplexe Teile des Materialfluss-Systems und somit eine rasche Abschätzung von Warteschlangenlängen und Auslastungen. Diese Vernetzung der Modelle der Transportsystem-Elemente geschieht in einem übergeordneten *kompositen Modell*. Dieses wird wegen seiner unterschiedlichen Teilsysteme hier auch als *Hybridmodell* bezeichnet. Auf die folgende Weise kann es erstellt werden:

### 1. Diskrete Simulationen...

... werden für jedes Transportsystem-Element bzw. für jede Klasse<sup>1</sup> von Elementen im Rahmen eines sinnvollen Parameterraums durchgeführt. Das Ergebnis eines einzelnen Simulationslaufs sind die zu dem benutzten Parametervektor gehörigen statistischen Kenngrößen, wie Warteschlangenquantile oder Auslastung des Elements. Die Anzahl der als Stützstellen benötigten Parameterkombinationen hängt von der konkreten Klasse ab und liegt bei den hier untersuchten Fällen bei 2000 bis 3000.

### 2. Neuronale Netze...

... werden für jede Klasse von Transportsystem-Elementen mit den Ergebnissen der Simulationsläufe trainiert. Das bedeutet, mit numerischen Methoden werden Funktionen gesucht, die für die benutzten Stützstellen die durch Simulation ermittelten Kenngrößen approximieren.

### 3. Ein Hybrides Modell...

... gestattet die strukturtreue Verknüpfung der Teilmodelle und berechnet für jedes Element die Auslastung und die Quantile der Warteschlangenlängen an den Eingängen. Dabei werden die im Voraus ermittelten numerischen Approximationen benutzt, um für die konkreten Eigenschaften eines Transportsystem-Elements (Parametervektor) bei einer konkreten logistischen Belastung die benötigten Kenngrößen zu berechnen.

---

<sup>1</sup>z.B. Regalbediengeräte, Einschleusungen, etc.

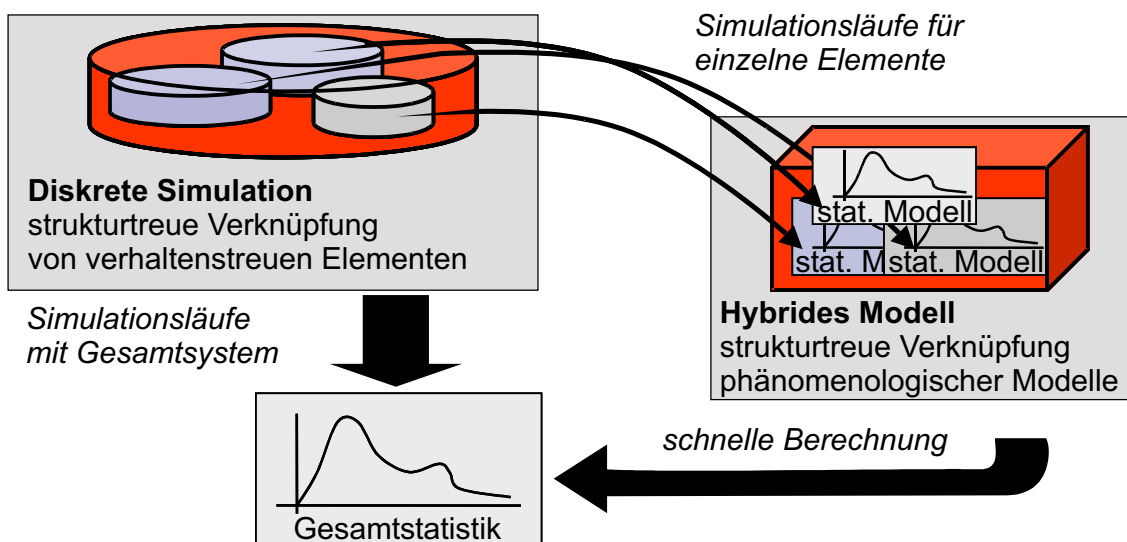
Die Schritte 1 und 2 sind extrem zeitaufwändig. Da dabei aber alle sinnvollen Parameterkonfigurationen der Transportsystem-Elemente berücksichtigt werden, reicht es aus, die Berechnungen einmalig durchzuführen. Die Komposition dieser einzelnen Teilmodelle zu einem hybriden Modell geschieht dann nach dem Baukastenprinzip.

Mit diesem Modell können folgende Materialfluss-Systeme mit folgenden Eigenschaften abgebildet werden (ausführliche Erläuterungen der Modellannahmen finden sich in Kapitel 4):

- Das System basiert auf Stetigförderern für den Stückguttransport.
- Blockierungen von Elementen gehören zu den Ausnahmen und sollen im normalen Systembetrieb vermieden werden (Blockierfreiheit).
- Die Steuerungen eines Elementes hängt nicht von Zuständen anderer Elemente ab (lokale Steuerungsregeln).

### Unterschied zur diskreten Simulation des gesamten Systems

Bei der diskreten Simulation werden Simulationsläufe mit kompletten, in der Regel aus einzelnen Bausteinen zusammengesetzten, Modellen durchgeführt (siehe Abbildung 1.1). Für eine konkrete Belastungssituation können statistische Untersuchungen vorgenommen werden. Auf dem neuen, hier vorgestellten Weg werden bereits im



**Abbildung 1.1:** „Statistik über Verknüpfungen“ oder „Verknüpfung von Statistiken“?

Vorfeld für jedes einzelne Transportsystem-Element für alle technisch sinnvollen Parameterkonfigurationen statistisch relevante Größen in einem numerischen Modell (neuronales Netz) abgebildet (vgl. Schritte 1 und 2, S. 3). Zur Modellierungszeit werden lediglich diese Teilmodelle strukturtreu verbunden und auf der Basis weiterer Parametervorgaben die interessierenden statistischen Größen des kompositen Modells berechnet (vgl. Schritt 3, ebd.).

Dieses Vorgehen bringt entscheidende Vorteile mit sich:

- Die Modellierung und Berechnung eines konkreten Systems benötigt im Vergleich mit einer diskreten Simulation so gut wie keine Zeit. Damit können sehr schnell unterschiedliche Situationen bewertet werden. Aufwändige Berechnungen sind nur *einmalig* nötig, um die numerischen Modelle für die einzelnen Modellklassen zu erstellen.
- Für jede Komponente werden bei der hier vorgestellten Methode alle sinnvollen Parameterkonfigurationen modelliert. Auch durch die Verknüpfung solcher Elemente bleiben die jeweiligen Parameter innerhalb der untersuchten Bereiche.

## 1.3 Aufbau der Arbeit

Das Kernstück der vorliegenden Arbeit ist die Entwicklung eines Modells, mit den Merkmalen, die im vorangegangenen Abschnitt skizziert wurden. Eine kurze Systematik der Modellierung wird deshalb in Kapitel 2 angeführt. Dort werden der aktuelle Stand der Forschung auf diesem Gebiet dargelegt, sowie Modelle für Materialflusssysteme vorgestellt und miteinander verglichen.

Für die numerische Approximation werden neuronale Netze benutzt. Eine kurze Erläuterung in Kapitel 3 soll den Leser<sup>2</sup> mit den grundlegenden Prinzipien der Arbeit mit neuronalen Netzen vertraut machen. Im Anschluss werden aktuelle Forschungsarbeiten diskutiert und die Einsatzmöglichkeiten neuronaler Netze in verwandten Gebieten vorgestellt.

Kapitel 4 bildet den Hauptteil dieser Arbeit. Dort wird Schritt für Schritt das neue Modell für ein Transportsystem-Element entwickelt. Die dabei notwendigen Modellannahmen werden ausführlich erläutert. Für grundlegende Materialfluss-Elemente werden die zugehörigen Modellklassen entwickelt und numerisch berechnet. Vertrauen in die einzelnen Modelle kann durch Verifikationen und Validierungen geschaffen werden.

Die Komposition der einzelnen Elemente zu einem hybriden Modell wird in Kapitel 5 beschrieben. Anhand von drei Beispielen kann die Leistungsfähigkeit des kompositen Modells demonstriert werden.

Abschließende Betrachtungen und Anregungen zu weiterführenden Forschungen finden sich in Kapitel 6.

Anhang A stellt kurz die im Rahmen dieser Arbeit benutzten Werkzeuge vor. In Anhang B finden sich Diagramme für die Modellverifikation und -validierung und in Anhang C ein Verzeichnis der verwendeten Symbole.

---

<sup>2</sup>Auch wenn hier ein generisches Maskulinum benutzt wird, sind in dieser Arbeit weibliche Personen gleichermaßen angesprochen.

# 2 Modellierung

Wichtiger Bestandteil dieser Arbeit ist ein Modellbildungsprozess. Um das Verständnis für die später in den Kapiteln 4 und 5 eingeführten Modelle zu erhöhen, werden hier zunächst die grundlegenden Begriffe „System“ und „Modell“ vorgestellt. Daran schließt sich eine kurze Systematisierung an.

Im dritten Abschnitt werden die gegenwärtig vorhandenen Möglichkeiten für die Modellierung von Materialfluss-Systemen diskutiert und miteinander verglichen.

## 2.1 Grundlagen

Eine Grundlage des Erkenntnisprozesses ist die Abstraktion, die Fähigkeit, Wesentliches von Irrelevantem und Systematisches von Zufälligem zu trennen. Ein Ergebnis eines Abstraktionsprozesses ist die Klassifizierung von Objekten auf Grund gewisser Merkmalsausprägungen. Dafür ist es notwendig, dass es eindeutig entscheidbar ist, ob ein Objekt zu einer Klasse gehört oder nicht (*Abgrenzung*). Aussagen, die dabei für die Objektklasse formuliert werden, müssen für jedes<sup>1</sup> ihrer Mitglieder zutreffen.

Ein *Objekt* ist dabei ein Ausschnitt der realen Welt oder selber wiederum eine Abstraktion (dh. eine Klasse von Objekten).

*Merkmale* oder *Attribute* sind Eigenschaften von Objekten. Diese Eigenschaften können für die Lebenszeit des Objektes fix oder variabel sein. Die Menge der Attribute eines Objektes  $o$  sei  $\mathcal{A}(o)$ . Mit  $\chi_t(\mathcal{A}(o))$  wird eine Wertbelegung der Attribute des Objektes  $o$  zur Zeit  $t$  bezeichnet. Der *Zustand* eines Objektes ist die Wertbelegung seiner, im Sinne der Abstraktion „relevanten“ Attribute zu einem konkreten Zeitpunkt.

**Bemerkung:** Nicht immer muss in der Abstraktion die „reale“ Zeit abgebildet werden. Im allgemeinen Falle betrachtet man dann die *Zeit* als eine nichtleere geordnete Indexmenge  $\mathcal{T}$ . Ein *Zeitpunkt*  $t$  ist dann  $t \in \mathcal{T}$ .

### 2.1.1 Systeme

Ein *System*  $\mathcal{S} = (\mathcal{O}, \mathcal{R}, \mathcal{A}, \chi)$  besteht aus einer Menge  $\mathcal{O}$  von Objekten, deren Attributen  $\{\mathcal{A}(o) \mid o \in \mathcal{O}\}$  und Wertbelegungen  $\chi_t(\mathcal{A}(.))$  und den Beziehungen  $\mathcal{R}$  zwischen den Objekten. Diese Beziehungen zwischen den Objekten sind Relationen zwischen den zugehörigen Attributen und immer in einem „zeitlichen“ Kontext zu

---

<sup>1</sup>In der Praxis, insbesondere in der anwendungsorientierten Forschung, wird diese Aussage unter Umständen auch abgeschwächt.

sehen: Ein Zustand eines Zeitpunktes  $t_2$  kann nicht Zustände zur Zeit  $t_1$  beeinflussen, wenn  $t_1 < t_2$ . (Dabei müssen nicht zu jedem Zeitpunkt  $t \in \mathcal{T}$  sämtliche Attributbelegungen bekannt oder erkennbar sein.)

Durch die Interaktion der Objekte und ggf. Wechselwirkungen nach außen wird eine ganzheitliche Aufgabe erfüllt. In der Literatur (vgl. [For70], [Sch70], [Law91], [Pag91], [Bos94]) wird oft ein „Zweck“ des Zusammenwirkens vorausgesetzt. Das erleichtert zwar das grundlegende Verständnis des Begriffes, impliziert aber planvolles Handeln. Aus mathematischer Herangehensweise (vgl. [Möl92]) beschränkt man sich bei der Definition des Systembegriffes auf Objekte und deren Beziehungen, wobei jedes Objekts direkt oder indirekt mit allen anderen in Relation steht. Die Sinnfälligkeit eines Systems leitet sich dann nicht aus der Definition sondern aus dem konkreten Fall ab.

Der *Systemzustand*  $\mathcal{Z}_t$  ist die Menge aller Einzelzustände seiner Elemente zu einem gegebenen Zeitpunkt.

Zur Veranschaulichung seien folgende wesentlichen Klassifizierungsmöglichkeiten von Systemen ohne Anspruch auf Vollständigkeit genannt (nach Möller [Möl92]):

- **Seinsberich: materiell/immateriell:**

Die untersuchten Objekte können Ausschnitte der Realität sein oder selber Abstraktionen.

- **Entstehung: natürlich/künstlich:**

Für materielle Systeme wird unterschieden, welche Ursprünge die Objekte eines Systems haben. Je nach Wahl des Beobachtungsstandpunktes kann ein künstliches System auch natürliche Objekte haben.

- **Äußere Wechselwirkung: offen/abgeschlossen:**

Wenn das System Wechselwirkungen mit seiner Umgebung erlaubt, nennt man es offen. Ansonsten abgeschlossen.

**Bemerkung:** Ein abgeschlossenes System bedeutet z.B. in der Thermodynamik, dass kein Wärmetransport zwischen dem System und seiner Umgebung stattfindet. Mit geeigneter Isolation ist dieser Zustand fast erreichbar. Das bedeutet jedoch nicht, dass es keine Wechselwirkungen zwischen dem System und seiner Umwelt gibt. Ungeachtet der thermischen Abgeschlossenheit existieren andere Beziehungen, wie etwa elektromagnetische, zwischen dem System und seiner Umwelt. Doch ein einfaches System der Thermodynamik abstrahiert von solchen Wechselwirkungen, sie werden nicht mit abgebildet.

- **Zeitliche Veränderlichkeit: statisch/dynamisch:**

Werden innerhalb eines Systems zeitliche Veränderungen zugelassen bzw. beschrieben, spricht man von einem dynamischen System. Spielt der Zeitverlauf keine Rolle im betrachteten System, dann nennt man es statisch. In diesem Fall entfällt die Bezugnahme der Wertbelegung  $\chi$  auf die Zeit.

- **Zeitverteilung der Zustände: diskret/kontinuierlich:**

Ist die Indexmenge der Systemzeit  $\mathcal{T}$  überabzählbar und zusammenhängend in  $\mathbb{R}$ , so heißt das System kontinuierlich. Für zwei beliebige Zeiten  $t_1, t_2 \in \mathcal{T}$

bedeutet das, jeder Zeitpunkt  $t_1 \leq t \leq t_2$  wird im Rahmen des Systems beschrieben. Ist hingegen  $\mathcal{T}$  abzählbar, heißt das System diskret.

**Bemerkung:** Mischformen „diskret-kontinuierlicher“ Art sind zum Beispiel Systeme, die Prozessverläufe beschreiben (kontinuierlich), die mit einem bestimmten Ereignis (diskret) ausgelöst werden.

#### - Beziehung zwischen den Objekten: deterministisch/stochastisch:

Gestattet das System die Abbildung zufälliger Zustandsänderungen, heißt es stochastisch, sonst deterministisch.

Materialfluss-Systeme sind meist komplex, sie bestehen oft aus abgrenzbaren, relativ autonomen Teilsystemen, deren Verhalten isoliert untersucht werden kann. Insbesondere werden einzelne Transportsystem-Elemente im Rahmen dieser Arbeit in separaten Modellen abgebildet und auf einer höheren Modellierungsstufe miteinander verknüpft.

**Bemerkung:** In der Technik findet man diese Modularität in Form von funktionsorientierten Baugruppen (etwa: Antriebsmaschine, Getriebe, Chassis, etc). Bei Organismen lassen sich solche Komponenten sowohl makroskopisch (Organsysteme, einzelne Organe) als auch mikroskopisch (Zellorganellen) ausmachen. Selbst bei der Untersuchung von geologischen, ökonomischen oder ökologischen Systemen können Teilsysteme einzeln untersucht werden. Für die Dokumentation von Betriebsprozessen ist die Beschreibung von Teilabläufen unabdingbar.

Für die Untersuchung eines komplexen Gesamtsystems werden also einerseits seine Teilsysteme isoliert betrachtet und andererseits deren Wechselwirkungen miteinander analysiert. Sind die einzelnen Teilsysteme hinreichend gut beschrieben, können sie fürderhin als *black boxes* angesehen werden, bei denen nur Ein- und Ausgänge interessieren. Diese modulare Betrachtungsweise gestattet es, die Dimension des Systemzustandes zu reduzieren. Erst durch den modularen Aufbau eines Transportsystems ist es möglich, in den Kapiteln 4 und 5 numerisch handhabbare Modelle zu entwickeln. Die Geschwindigkeit der Berechnung und die Allgemeinheit des dort vorgestellten Hybridmodells beruhen auf seiner Modularität.

### 2.1.2 Modelle

Aussagen über bestimmte Eigenschaften eines Systems können durch direkte Beobachtung gewonnen werden. Doch nicht immer ist dies praktikabel oder möglich. Gründe dafür können zum Beispiel sein:

- die Zugänglichkeit von Daten (nicht alle Zustände sind immer erkennbar),
- die Prozessgeschwindigkeit des Systems ist zu klein (z.B. bei Ökosystemen) oder zu groß (etwa bei der Explosion in einem Verbrennungsmotor),
- das interessierende System gestattet keine Experimente (z.B. aus physikalischen, technologischen, ökonomischen oder ethischen Gründen).

Für ein gegebenes System  $\mathcal{S} = (\mathcal{O}, \mathcal{R}, \mathcal{A}, \chi)$  und ausgewählte Teilmengen  $\tilde{\mathcal{O}} \subseteq \mathcal{O}$ ,  $\tilde{\mathcal{R}} \subseteq \mathcal{R}$  und  $\tilde{\mathcal{A}} \subseteq \mathcal{A}$  von interessierenden Objekten, Beziehungen und Attributen ist ein *Modell* ein System  $\mathcal{S}' = (\mathcal{O}', \mathcal{R}', \mathcal{A}', \chi')$ , das durch eine Abbildung  $\mathbf{M}$  (Modellierung) mit dem ursprünglichen System in Beziehung steht:

$$\begin{array}{ccc} \mathcal{S} & \xrightarrow{\mathbf{M}} & \mathcal{S}' \\ \tilde{\mathcal{O}} & \mapsto & \mathcal{O}' = \mathbf{M}\tilde{\mathcal{O}} \\ \tilde{\mathcal{R}} & \mapsto & \mathcal{R}' = \mathbf{M}\tilde{\mathcal{R}} \\ \tilde{\mathcal{A}} & \mapsto & \mathcal{A}' = \mathbf{M}\tilde{\mathcal{A}} \\ \chi|_{\tilde{\mathcal{A}}} & \mapsto & \chi' = \mathbf{M}\chi|_{\tilde{\mathcal{A}}} \end{array}$$

Durch die Modellierung werden nur die ausgewählten Objekte mit den dazugehörigen Beziehungen und Attributen abgebildet. Damit können je nach Blickwinkel für ein System verschiedene Modelle benutzt werden. Ein Modell ist immer eine Form der Vereinfachung des interessierenden Systems unter einem bestimmten Aspekt.

An einem geeigneten Modell können Experimente und Analysen besser durchgeführt werden bzw. sind überhaupt erst möglich. Die Auswertung der Ergebnisse kann nun auf zwei Wegen erfolgen: Wenn die Modellierung die wesentlichen Zusammenhänge erhält und wenn es eine Abbildung gibt, die die relevanten Zustände semantisch oder funktionell korrekt vom Modell zurück in das originale System überträgt, so sind die Ergebnisse der Modellierung direkt nutzbar<sup>2</sup>. Ist im anderen Falle keine Rückübertragung möglich, können die Resultate der Modellierung als *zusätzliche* Daten des Systems zu dessen Beschreibung herangezogen werden. Dies ist insbesondere dann der Fall, wenn Zusammenhänge durch die Modellierung extrem vereinfacht werden.

Das folgende kurze Beispiel sei zur Illustration angeführt: Gegeben sei ein System, das aus einer Bearbeitungsmaschine, dem vorgelagerten Warteraum und den zu bearbeitenden Werkstücken besteht. Die Zustände des Warteraums (Anzahl der wartenden Objekte) und der Maschine (belegt oder frei) ändern sich wie folgt: Ist die Maschine frei und ein neues Werkstück trifft ein, wird die Maschine für eine konstante Zeit (Bedienzeit  $t_b$ ) belegt. Ist die Maschine belegt und ein neues Werkstück trifft ein, erhöht sich die Anzahl der wartenden Objekte um Eins. Beendet die Maschine die Bearbeitung eines Objektes und im Warteraum warten Werkstücke, wird die Anzahl der Wartenden um 1 verringert, die Maschine für  $t_b$  belegt. Ist der Warteraum leer, dann ist die Maschine nach der Bearbeitung des Werkstücks wieder frei. Dann sind folgende Betrachtungsweisen möglich:

- a) Wenn die Werkstücke von außen in das System eintreten, hat man es mit einem offenem System zu tun. Modelle für dieses System wären etwa ein Zustandsgraph, oder ein Softwareprogramm, das genau dieses Verhalten abbildet. Die Übertragung vom Modell auf das ursprüngliche System ist ohne Weiteres möglich.
- b) Wird im System noch die mittlere Zwischenankunftszeit eintreffender Werkstücke berücksichtigt, kann ein einfaches M/D/1-Bediensystem zur Modellierung des

<sup>2</sup>Zur Korrektheit der Modellierung s.a. Kapitel 2.2.5.

Problems aufgestellt werden. Damit sind statistische Aussagen über die Anzahl der wartenden Werkstücke möglich.<sup>3</sup> Wenn im betrachteten System die statistische Verteilung der Ankunftszeiten von der im Modell angenommenen Exponentialverteilung abweicht, sind diese Aussagen des Modells nicht mehr direkt auf das System übertragbar. Sie sind jedoch unter Umständen gute Abschätzungen für die entsprechenden Eigenschaften des Originals und können für die Bewertung des Systems genutzt werden.

## 2.2 Systematik

In Rahmen dieser Arbeit wird eine neue Methode der Modellierung von Transportsystemen entwickelt. Um sie im Umfeld der Modellierung einordnen zu können, wird an dieser Stelle eine kurze Systematik von Modellen vorgestellt.

### 2.2.1 Modellzweck

Ausschlaggebend für die Auswahl eines Modelles zur Beschreibung eines Systems ist der verfolgte Zweck. Zwei grundlegende Fragen stehen dabei im Vordergrund:

- a) Wie verhält sich das System unter gewissen (konkreten) Bedingungen?  
Hierbei geht es im Wesentlichen um die Vorhersage von Systemzuständen, z.B. die Reaktion des Systems auf eine besondere Belastungssituation.
- b) In welchem Rahmen bewegen sich Systemzustände?  
Die Gewährleistung von Systemzuständen, z.B. der Nachweis der Leistung eines technischen Systems, steht hier im Mittelpunkt.

Bei der Modellierung ist stets eine Balance zwischen Allgemeingültigkeit, Exaktheit und Handhabbarkeit zu finden: Aussagen von zu allgemeinen Modellen lassen unter Umständen keine Antwort auf spezielle Fragestellungen zu oder sind zu ungenau. Detaillierte Modelle sind zwar für lokale Probleme sehr gut geeignet, für sehr komplexe Systeme mag es aber angemessen erscheinen, unter Zuhilfenahme einer gegebenen Modularität, den Modellierungsfokus – die Sichtweise auf das System – zu verändern.

Die in den Kapiteln 4 und 5 beschriebenen Modelle dienen dazu, ausgehend vom konkreten Transportsystem eine schnelle Abschätzung über gewisse statistische Kenngrößen seiner Elemente zu treffen. Das komposite hybride Modell bildet die Struktur des Transportsystems ab; dessen einzelne Elemente werden mit Hilfe von neuronalen Netzen phänomenologisch modelliert.

---

<sup>3</sup>Einzelheiten zur Modellierung von Materialfluss-Systemen sind in Abschnitt 2.3 zu finden.



### 2.2.2 Art der Modellierung

Es existieren zwei grundlegend verschiedene Möglichkeiten der Modellierung: Nachbilden der Systemstruktur oder Nachbilden von Systemmerkmalen. Dazwischen sind – vor allem in der Praxis – Mischformen gebräuchlich.

- **Erklären von Merkmalen - Strukturelle Modellierung**

Der Kern dieser Herangehensweise besteht im Erkennen und Beschreiben von kausalen und funktionalen Zusammenhängen des Systems. Beobachtungen des Verhaltens des realen Systems sind höchstens zur Verifikation erforderlich. Ein verhaltens erklärendes Modell erlaubt die Vorhersage von Eigenschaften des Systems, die bislang noch nicht beobachtet wurden.

- **Beschreiben von Merkmalen - Phänomenologische Modellierung**

Unabhängig von den realen Zusammenhängen im betrachteten System werden Merkmale des Systems nachgebildet. Das Modell wird allein auf der Basis von Beobachtungen des realen Systems entwickelt. Die strikte Gültigkeit bleibt beschränkt auf bereits erfolgte Beobachtungen.

- **Mischformen**

Wenn die Struktur des Systems prinzipiell weitgehend bekannt ist, werden Beobachtungen des realen Systems benutzt, um bislang unbekannte Modellparameter so anzupassen, dass das Modell die beobachteten Merkmale möglichst genau abbildet.

Im Bereich der Naturwissenschaften werden eher analytische Modelle benutzt um gewisse Merkmale zu *erklären*. In den angewandten Wissenschaften werden Modelle bevorzugt, die Prozesse oder Eigenschaften von Objekten handhabbar beschreiben können.

**Bemerkung:** Ausgehend von materiellen Objekten wurden innerhalb der klassischen Physik Modelle („Naturgesetze“) formuliert, die die Realität hinreichend gut widerspiegeln.<sup>4</sup> Auf diesen Grundlagen fußt die klassische Mechanik und benutzt sie als Axiome bei der Modellierung von Mechanismen und komplexen Maschinensystemen. Gleichzeitig wird in der modernen Physik der Fokus verkleinert. Das erlaubt es, bis dato phänomenologische Modelle auf einer anderen Stufe durch allgemeinere, analytische Modelle (Quanten- und Stringtheorie) zu ersetzen, ohne dass dadurch die Gültigkeit im makroskopischen Maßstab beeinflusst wird.

Der in dieser Arbeit vorgestellte Ansatz bildet statistische Merkmale von Transportsystem-Elementen ab. Es handelt sich dabei um eine phänomenologische Modellierung (Kapitel 4) bzw. eine Mischform (Kapitel 5).

---

<sup>4</sup>Die Gravitationskraft  $F$  zwischen zwei Objekten hängt linear von ihren Massen  $m_1, m_2$  und quadratisch von dem Reziproken ihres Abstandes  $r$  ab:  $F = \gamma \frac{m_1 m_2}{r^2}$ . Die Gravitationskonstante  $\gamma$  wurde experimentell ermittelt. Dieses Modell stellt also eine Mischform im obigen Sinne dar. Durch die Einbettung in eine ganzheitliche „Welttheorie“ kann sie strukturell beschrieben werden.

### 2.2.3 Gegenstand der Modellierung

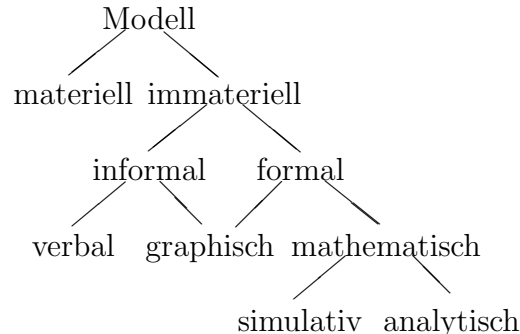
Die Systematik der Modellierung umfasst auch den Gegenstand der Modellierung. Das betrachtete originale System kann entsprechend der Klassifikation auf S.7f. eingeordnet werden.

**Bemerkung:** Die in den Kapiteln 4 und 5 beschriebenen Modelle können damit folgendermaßen charakterisiert werden:

- Obwohl ausschließlich statistische Größen modelliert werden, haben jene doch einen materiellen Hintergrund.
- Es werden abgeschlossene künstliche Systeme beschrieben. Das bedeutet insbesondere, die Beschreibungen der Systemeingänge sind Bestandteil des Systems.
- Auch wenn Transportsysteme dynamisch und stochastisch sind, handelt es sich bei den vorgestellten Modellen um statische und deterministische. Es werden nicht einzelne Ereignisse abgebildet sondern statistische Kenngrößen.

### 2.2.4 Methoden

Eine grobe Untergliederung nach der Art und Weise der Modellierung findet sich unter anderem bei Page [Pag91]:



**Abbildung 2.1:** Systematik von Modellen

Unter einem informalen Modell ist eine Beschreibung etwa in Umgangssprache oder in Form von graphischen Übersichten zu verstehen. Formale graphische Darstellungen sind zum Beispiel Bildgraphen oder Flussdiagramme. Die Simulation zählt im Allgemeinen zu den mathematischen Modellen, aus ingenieurtechnischer Sicht ist dabei der Fokus auf die Simulation technischer Systeme gerichtet.<sup>5</sup>

Die analytische Modellierung beinhaltet die gesamte Palette der Mathematik. Stellvertretend seien partielle Differentialgleichungen zur Modellierung kontinuierlicher Systeme, Graphen- und Algorithmentheorie zur Analyse von Softwaresystemen, Algebra für die Abbildung von Quanteneffekten oder deskriptive Statistik zur phänomenologischen Modellierung genannt.

<sup>5</sup>Allerdings gibt es auch verbale Simulationen, wie etwa Rollenspiele im Bereich der Geistes- und Sozialwissenschaften.

### 2.2.5 Gültigkeit eines Modells

In welchem Maße ein Modell das originale System widerspiegelt, kann im Wesentlichen unter vier Blickwinkeln betrachtet werden - jedoch ist immer die Bezugnahme zum Modellzweck entscheidend [Bos94]:

- **Verhaltensgültigkeit**

Das qualitative Verhalten des Modells entspricht dem des Originals.

- **Strukturgültigkeit**

Die interessierende Struktur des originalen Systems entspricht der des Modells.

- **Empirische Gültigkeit**

Die beobachteten Merkmale des Systems werden durch das Modell hinreichend exakt abgebildet. Andere Merkmale des Modells sind konsistent und plausibel.

- **Anwendungsgültigkeit**

Das Modell entspricht dem Modellzweck und den Anforderungen des Anwenders.

**Validierung** ist der Versuch der Begründung, dass die Anwendung eines Modells *sinnvoll* ist [Rob97]. Verschiedenen Schritte dienen dabei dazu, die Zuverlässigkeit und den Gültigkeitsbereich des Modells zu testen.

Ein Bestandteil des Validierungsprozesses ist die **Verifikation** als Nachweis der Korrektheit des Modells. Dabei geht es primär um die innere Stimmigkeit und Fehlerfreiheit *innerhalb* des Modells.

Je komplexer ein Modell ist, umso schwerer ist es, zu zeigen, dass das Modell korrekt rechnet. In den meisten Fällen ist ein „Beweis“ unmöglich. Insbesondere, wenn das Ergebnis eines Modells fehlerbehaftet ist, geht es bei der Verifikation vorrangig darum, dass sich Entwickler und Nutzer des Modells anhand von Tests von der Korrektheit des Modells überzeugen.

In stärkerem Maße gilt das für die Validierung. Sie ist meist nur durch den praktischen Einsatz des Modells und den Vergleich seiner Ergebnisse mit anderen Daten möglich.

Das Resultat von Verifikation und Validierung ist nicht entweder Ablehnung oder Akzeptanz eines Modells. Meist führen fehlgeschlagene Versuche zu einer Weiterentwicklung des Modells. In der Regel wird erst dann, wenn Nachbesserungen unmöglich sind, das Modell verworfen.

Eine sehr schöne und umfassende Monographie über Modellierung und Gültigkeit von Modellen findet man bei Penrose [Pen89]. Breiten Raum nimmt dort auch die Diskussion über künstliche Intelligenz ein.

Bei den Modellen, die in den Kapiteln 4 und 5 vorgestellt werden, tritt der hybride Charakter deutlich zutage: Für die Modellierung der einzelnen Transportsystem-Elemente geht es um den Nachweis der empirischen Gültigkeit. Bei der Vernetzung der Elemente ist die Struktur des Systems korrekt abzubilden. Für das gesamte hybride Modell ist jedoch seine Anwendungsgültigkeit zu erbringen.

## 2.3 Modellierung von Materialfluss-Systemen

Für die Untersuchung von Transportsystemen spielt die Modellierung von Wartebereichen eine wesentliche Rolle. Als statistische Maßzahl wird hier dafür das *Quantil* benutzt. Im folgenden Abschnitt wird erläutert, mit welchem Quantilbegriff im Rahmen dieser Arbeit operiert wird.

Im Anschluß daran werden die drei wesentlichen Möglichkeiten für die Modellierung von Materialfluss-Systemen kurz vorgestellt und miteinander verglichen.

### 2.3.1 Quantile

An dieser Stelle wird zunächst der klassische Quantilbegriff vorgestellt. Die sich aus seinem diskreten Charakter ergebenden Probleme werden skizziert und durch eine stetige Formulierung des Quantilbegriffs aufgehoben.

Sei  $X$  eine Zufallsgröße, dann heißt ein Wert  $q_p$ , der den Ungleichungen

$$P(X \leq q_p) \geq p \quad \text{und} \quad P(X \geq q_p) \geq 1 - p \quad (2.1)$$

genügt, *Quantil der Ordnung  $p$* . Ist  $X$  mit  $F_X(\cdot)$  stetig verteilt so sind die Quantile  $p$ -ter Ordnung solche Werte von  $x$ , für die

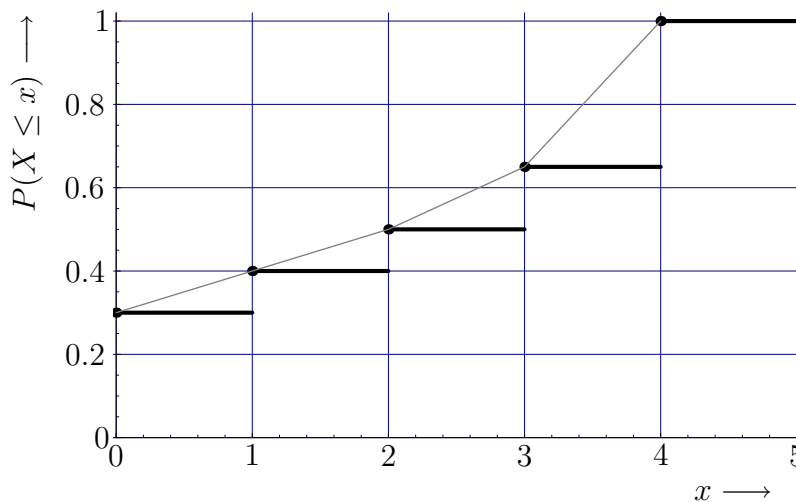
$$F_X(x) = p \quad (2.2)$$

gilt [Fis76].

Nach dieser Definition gibt es nicht notwendig für jede Wahrscheinlichkeit  $0 \leq p \leq 1$  ein zugehöriges Quantil. Als Beispiel sollen die folgende diskreten Wahrscheinlichkeiten  $p_i = P(X = x_i)$  betrachtet werden:

$x_i$	0	1	2	3	4
$p_i$	0.3	0.1	0.1	0.15	0.35

Für die in Abbildung 2.2 skizzierte zugehörige diskrete Verteilungsfunktion der Zufallsgröße  $X$  gibt es lediglich Quantile der Ordnungen 0, 0.3, 0.4, 0.65 und 1.0.



**Abbildung 2.2:** Verteilungsfunktion  $F_X(x)$  einer diskreten Zufallsgröße  $X$  (fett) und ihre lineare Interpolation  $\tilde{F}(x)$  für  $0 \leq x$

Was hier interessiert, ist eine Größe, die sagt, wie lang eine Warteschlange mit einer gewissen Wahrscheinlichkeit  $p$  höchstens ist. Also als Veränderung der Definition: die kleinste Zahl  $q_p$ , für die die linke Seite aus (2.1) gilt. Für das dargestellte Beispiel wäre das so definierte Quantil der Ordnung  $p = 0.7$  erst  $x = 4$ , da  $P(X \leq 3) = 0.65 < p$ .

**Bemerkung:** Sicherlich ist letztendlich eine konkrete Anzahl von Stauplätzen vor einem Transportsystem-Element festzulegen. Ein Quantil der Warteschlangenlänge ist jedoch nicht das einzige Kriterium dafür – es soll den Planer lediglich bei seiner Entscheidung unterstützen. Aus diesem Grund scheint es angebracht, hier von diskreten Größen abzugehen.

Durch eine lineare Interpolation erhält man eine stetige Formulierung: Sei  $X$  eine diskrete Zufallsgröße mit den Wahrscheinlichkeiten  $P(x_i) = p_i$  für  $i = 1, 2, \dots$  und  $x_i < x_{i+1}$ . Die Funktion  $\tilde{F}(\cdot)$  sei folgendermaßen definiert:

$$\tilde{F}(t) = \begin{cases} 0, & \text{falls } t < x_1, \\ \sum_{j=1}^i p_j + p_{i+1} \cdot \frac{t-x_i}{x_{i+1}-x_i}, & \text{falls } x_i \leq t < x_{i+1} \\ 1, & \text{sonst.} \end{cases} \quad (2.3)$$

Für  $p_1 \leq t$  interpoliert  $\tilde{F}(t)$  die durch die Verteilungsfunktion  $F(\cdot)$  bestimmten Stützstellen  $(x_i, F(x_i))_{i=1,2,\dots}$  linear (siehe Abbildung 2.2, dünne Linie).

Abweichend von der oben angeführten Definition wird eine Zahl  $Q_p$  mit  $p = \tilde{F}(Q_p)$  als ein *verstetigtes Quantil* der Ordnung  $p$  bezeichnet. Im Rahmen dieser Arbeit wird bis auf eine Ausnahme (auf S. 47) das verstetigte Quantil  $Q_p$  statt  $q_p$  verwendet. Für  $Q_p$  wird deshalb hier die verkürzende Bezeichnung „Quantil“ benutzt.<sup>6</sup>

Für obiges Beispiel (Abbildung 2.2) hat das (verstetigte) Quantil der Ordnung 0.7 den Wert 3.14.

### 2.3.2 Statische Berechnungsmodelle

Statische Berechnungen beruhen im Allgemeinen auf der Auswertung von Mittelwerten von Systemkenngrößen.

Sei  $T$  ein festes Zeitintervall (z.B.  $T = 1$  h) und  $T_B$  die Summe der einzelnen Zeiten, die ein Element im Verlauf der Zeit  $T$  mit der Bedienung von Forderungen beschäftigt ist. Als *Auslastung*  $\varrho$  wird dann der Quotient

$$\varrho = \frac{T_B}{T} \quad (2.4)$$

bezeichnet.  $\varrho = 0$  bedeutet, das Element wird nicht benutzt.  $\varrho = 1$  heißt, das Element hat keine Leerlaufzeit.

Die *Streckenlast* eines Förderers ist die mittlere Anzahl der Transporteinheiten, die in der Zeit  $T$  diesen Förderer passieren. Üblich ist es, die Streckenlast über die Anzahl der Bewegungen pro Stunde anzugeben.

<sup>6</sup>Als Schreibweise wird für das 90%-Quantil  $Q_{90}$  verwendet. Falls die betrachtete Ordnung des Quantils klar ist, wird auch auf ihre Angabe als Index verzichtet.

Wenn  $X_i$  ( $i = 1, \dots, N$ ) ausgewählte Elemente eines Transportsystems sind (Bearbeitungsmaschinen, Ein- oder Ausgangsstationen etc.), so sei  $m_{ij}$  die mittlere Anzahl der Bewegungen von Transporteinheiten von  $X_i$  nach  $X_j$  in der Zeit  $T$ . Dann wird die quadratische Matrix mit den Elementen  $m_{ij}$  als *Transportmatrix* dieses Systems bezeichnet:

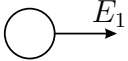
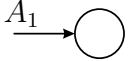
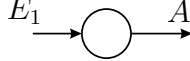
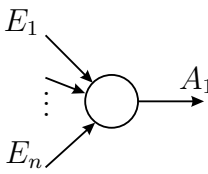
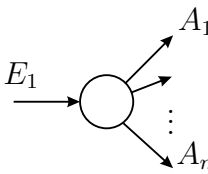
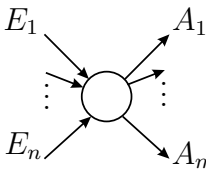
$$M = (m_{ij})_{(i,j=1,\dots,N)}. \quad (2.5)$$

Sei  $t_{ij}$  die mittlere Zeit, die eine Transporteinheit für den Weg von  $X_i$  nach  $X_j$  benötigt. Die Matrix

$$W = (t_{ij})_{(i,j=1,\dots,N)}. \quad (2.6)$$

wird als *Wegzeitmatrix* definiert.

Gudehus [Gud99] ordnet einem Transportsystem-Element mit  $n$  Eingängen  $E_1, \dots, E_n$  und  $m$  Ausgängen  $A_1, \dots, A_m$  den *Typ*  $(n, m)$  zu (siehe Tabelle 2.1). Die Summe  $n + m$  bezeichnet er als *Ordnung* des Elements.

		
<b>Quelle</b> $(0,1)$	<b>Senke</b> $(1,0)$	<b>Bedienstation</b> $(1,1)$
		
<b>Zusammenführung</b> $(n,1)$	<b>Verzweigung</b> $(1,m)$	<b>Kreuzung</b> $(n, m)$

**Tabelle 2.1:** Typen von Transportsystem-Elementen nach Gudehus

Diese Systematisierung wird benutzt, um einen Graphen eines Transportsystems aufzustellen. Typische statische Berechnungen sind die folgenden:

- Basierend auf dem Graphen des Transportsystems kann mit Hilfe der Transportmatrix die mittlere Belastung für jedes Element des Systems berechnet werden. Ist die mittlere Bedienzeit des Elements bekannt, kann seine Auslastung nach Gleichung (2.4) berechnet werden.
- Für Fahrzeugsysteme kann mit den Transport- und Wegzeitmatrizen abgeschätzt werden, wie viele Fahrzeuge für die Erfüllung der Transportaufgaben benötigt werden. Dabei müssen außer den Wegzeiten für Transporte von Gütern auch Leerfahrten der Fahrzeuge berücksichtigt werden: Mit einem Leerfahrzeug-Ausgleich lässt sich eine untere Abschätzung für die benötigten Leerfahrten berechnen.
- Durch die Berechnungen von Grenzleistungen einzelner Transportsystem-Elemente kann ein Leistungsnachweis für kritische Stellen erbracht werden.

Diesen Modellen ist gemein, dass mit Hilfe weniger Berechnungen ein Wert berechnet werden kann, der unter den getroffenen Modellannahmen exakt ist. Diese

Annahmen fordern jedoch meist, dass die Gutströme oder/und die Bedienzeitfunktionen der betrachteten Elemente markovsch sind [Gud75a]. Meistens weicht die Realität jedoch stark von den Annahmen ab, so dass die Berechnungsergebnisse nicht ohne Weiteres übernommen werden können (vgl. S. 9).

Insbesondere, wenn durch diese Methoden möglicherweise kritische Bereiche erkannt werden, ist es oft sinnvoll, Detailuntersuchungen, beispielsweise mit diskreten Simulationen, durchzuführen.

### 2.3.3 Wartesysteme

Mit der Untersuchung statistischer Prozesse in Fertigungssystemen beschäftigt man sich im Rahmen der Warteschlangentheorie (*queueing theory*) oder Theorie der Wartesysteme [Fer64]. In Abhängigkeit von den Ankunfts- und Bedienprozessen werden dabei auf analytischem Weg die resultierenden Warteschlangenlängen berechnet (z.B. [Gud75b], [Gud99]).

Als *Zustand* eines Wartesystems wird die Länge der Warteschlange vor einer Bedienstation definiert. Für markovsche („gedächtnislose“) Ankunftsprozesse ist die Wahrscheinlichkeit dafür, dass im folgenden Zeitintervall  $\Delta t$  eine Forderung eintrifft, einzig von  $\Delta t$  abhängig. Damit ist es für Bediensysteme mit markovschen Ankunfts- und Bedienprozessen möglich, einen Zustandsgraphen anzugeben, der einzig von den Ankunfts- und Bedienraten des Systems abhängt.

Eine *Blockierung* eines Transportsystem-Elements liegt dann vor, wenn es durch ein nachfolgende Element daran gehindert wird, die Bedienung einer Transporteinheit zu beginnen bzw. zu beenden. Gründe für eine Blockierung können Ausfälle oder unzureichende Stauräume sein.

Die Betrachtung von Warteschlangennetzen [Alt87] ermöglicht es, Blockierungen vor Bedienstationen und deren Auswirkungen in komplexen Strukturen zu untersuchen. Nach geschickten Dekompositionen muss dabei für jedes Element der Zustandsgraph des lokalen Wartesystems ausgewertet werden. Meist werden dafür die Ankunfts- und Bedienzeiten als exponentialverteilt angenommen [Fur92]. Der Einsatz von mehrphasigen Verteilungsfunktionen ermöglicht die Berücksichtigung von statistischen Momenten höherer Ordnung, ohne die Markov-Eigenschaft dafür aufgeben zu müssen [Ral98].

Für fördertechnische Elemente führt ein Ansatz der Warteschlangentheorie leider meist sehr schnell in eine Sackgasse. Grund dafür ist, dass die Bedienzeit von Transportsystem-Elementen sehr stark von ihrem aktuellen Zustand abhängt.

Als Beispiel wird nun ein (Rollen-)Hubtisch betrachtet. Je nachdem, in welcher Position sich die Hubvorrichtung befindet, können Transporteinheiten auf Rollen- oder auf Kettenbasis transferiert werden. Das bedeutet, die Bedienzeit für eine Transporteinheit hängt davon ab, in welche Richtung der vorhergehende Transfer stattgefunden hat. Denn diese bestimmt, ob vor der Übernahme der Transporteinheit eine Positionierung der Hubvorrichtung notwendig ist. Kommen hier noch Regeln für die Auswahl des nächsten Transportauftrages oder für die Positionierung der Hubvorrichtung nach der Beendigung eines Auftrages hinzu, sind die Bedienzeitfunktionen nur schwer explizit formulierbar.

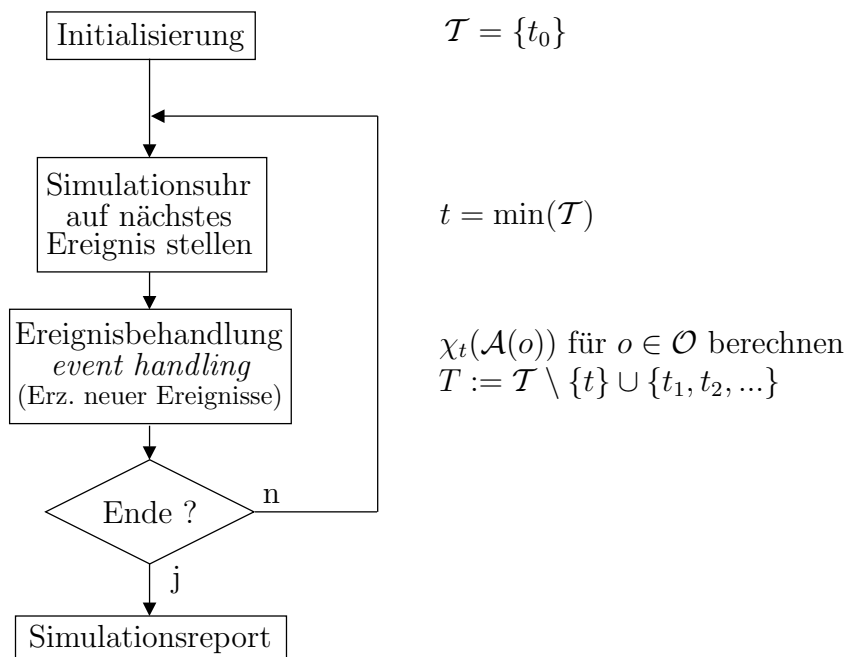
Das Aufstellen des Zustandsgraphen für die Warteräume vor dem Hubtisch ist hier ungleich aufwändiger, da hier die Zustände beider Warteräume und die Position des Hubtisches berücksichtigt werden müssen. Eine explizite Berechnung ist zwar unter Umständen möglich, aber die Ergebnisse sind nicht handhabbar<sup>7</sup>.

Unter der Annahme exponentialverteilter Ankunfts- und Bedienzeiten erhält man jedoch auf einfachem Weg Mittelwerte für die Längen der Warteschlangen und der Wartezeiten [Gud75b]. Diese Ergebnisse können jedoch nur sehr bedingt zur Dimensionierung des Systems benutzt werden: Die mittlere Warteschlangenlänge eignet sich nicht dafür, eine Aussage über den Stauplatzbedarf für eine hohe statistische Sicherheit abzuleiten.

### 2.3.4 Diskrete Simulation

Wichtigstes Hilfsmittel zur detaillierten Modellierung von Materialfluss-Systemen ist die (ereignis-)diskrete Simulation. Deren Konzept soll hier kurz umrissen werden.

Als *Ereignis* wird ein Auslöser einer Zustandsänderung betrachtet. Ereignisse in einem Materialfluss-System sind zum Beispiel Ankunft einer Transporteinheit oder die Freigabe einer Maschine oder Transportvorrichtung. Bei der diskreten Simulation (*discrete event simulation*) wird die Zeit als abzählbare Indexmenge  $\mathcal{T}$  abgebildet (vgl. 2.1). Jedem Ereignis entspricht eineindeutig ein Zeitpunkt dieser Menge.



**Abbildung 2.3:** Grundstruktur der diskreten Simulation

Nach der Initialisierung der Attribute der modellierten Objekte wird die Simulation mit  $\mathcal{T} = \{t_0\}$  begonnen. Zum jeweils nächsten Ereigniszeitpunkt  $t \in \mathcal{T}$  werden im Rahmen der Algorithmen der „Ereignisbehandlung“ die Attribute von

<sup>7</sup>Selbst für vergleichsweise einfache Materialfluss-Elemente muss der analytische Weg aus Komplexitätsgründen verlassen werden [Mar98a], [Kre01].



Objekten  $o$  des Modells neu berechnet. Dabei werden gegebenenfalls weitere Ereignisse zur Indexmenge  $\mathcal{T}$  hinzugefügt.

Durch diese algorithmische Modellierung ist es möglich, Steuerungen des realen Systems so detailliert wie nötig nachzubilden. Deshalb kann hier fast eine beliebige Genauigkeit erreicht werden [Pag91]. Der Aufwand für den Aufbau des Simulationsmodells ist jedoch unter Umständen sehr groß. Aber die Simulation gestattet einen tiefen Einblick in die ablaufenden Prozesse („*Simulation is insight, not numbers.*“ [Ban98]). Da ein Simulationslauf meist von vielen Parametern beeinflusst wird, sind funktionale Aussagen über deren Einfluss auf das Simulationsergebnis selten möglich oder angestrebt. Schwerpunkte für den Einsatz diskreter Simulationen sind vielmehr die Verifizierung von Steuerungsalgorithmen, etwa zur Vermeidung von Staus, und der Leitungsnachweis des Gesamtsystems für konkrete Konfigurationen [Ban01].

Vermehrt kann auch beobachtet werden, dass diskrete Simulationen eingesetzt werden, um ansonsten unzugängliche Daten eines laufenden System zu erhalten: Angeregt durch Ereignisse des realen Systems ist die Simulation in der Lage, für jede Zeit den genauen Systemzustand zu berechnen. Dadurch können diskrete Simulationen darüber hinaus für Prognosen zur Echtzeit eingesetzt werden.

### 2.3.5 Vergleich von Modellierungsarten

In einer Übersicht (Tabelle 2.2) sollen an dieser Stelle wesentliche Charakteristika der gegenwärtigen Modellierungsarten von Materialfluss-Systemen verglichen werden.

	Aufwand der Entwicklung		Aufwand des Einsatzes		Allgemeingültigkeit	Detailtreue	Einsatz
Statische Modelle	sehr klein	klein	groß	gering			Grobplanung, Grenzleistungen einzelner Elemente
Wartesysteme	sehr groß	klein	groß	hoch			Dimensionierung von Stauräumen
Diskrete Simulation	groß	groß	klein	sehr hoch			Test der Steuerung, Leistungsnachweis

**Tabelle 2.2:** Vergleich von Modellierungsarten

### 2.3.6 Metamodelle

Wie in Abschnitt 2.1 angedeutet, können auch Modelle Objekte einer Modellierung sein. Als *Metamodell* wird solch ein Modell von Modellen bezeichnet.

Meist werden numerische Metamodelle von Transportsystem-Elementen betrachtet: Für eine ganze Klasse von Elementen werden gewisse Eigenschaften mit einem Metamodell prognostiziert. Sei  $\boldsymbol{\pi} = (\pi_1, \pi_1, \pi_2, \dots, \pi_n)$  ein Parametervektor zur Beschreibung eines Repräsentanten der modellierten Klasse. Die zu modellierende Eigenschaft sei  $y = y(\boldsymbol{\pi})$ .

Sie ist in der Regel nur als Berechnungs- oder Simulationsergebnis für eine gewissen Menge  $\Pi$  von Parametertupeln (Stützstellen) gegeben. Dann bezeichnet man

$$\{(\boldsymbol{\pi}, y(\boldsymbol{\pi})) \mid \boldsymbol{\pi} \in \mathbb{R}^n\} \subset \mathbb{R}^{n+1} \quad (2.7)$$

als Hyperfläche (Wirkungsfläche) der Funktion (Eigenschaft)  $y$ .

Durch eine Funktion  $\tilde{y} : \mathbb{R}^n \rightarrow \mathbb{R}$  soll diese Hyperfläche an allen bekannten Stellen  $\boldsymbol{\pi}$  approximiert werden:

$$\forall \boldsymbol{\pi} \in \Pi : y(\boldsymbol{\pi}) \approx \tilde{y}(\boldsymbol{\pi}). \quad (2.8)$$

Ein klassischer Ansatz dafür ist, Polynome niedriger Ordnung zu benutzen. Durch den Einsatz von genetischen Algorithmen können in [Sch00a] für Hyperflächen geringer Krümmung auch Polynome mit höheren Koeffizienten erfolgreich für die Modellierung ausgewählter Bediensysteme benutzt werden. – Auf die Approximation von Hyperflächen mit neuronalen Netzen wird im nächsten Kapitel ausführlich eingegangen.

Metamodelle können sowohl für die statische Modellierung als auch als Ersatz für Teilsysteme in der diskreten Simulation benutzt werden. In [Völ02] werden in ihrer Genauigkeit (und ihrem Aufwand) reduzierte Modelle für Produktionsbereiche eingesetzt, um damit eine schnelle Termin- und Kapazitätsplanung zu ermöglichen.

Metamodelle wurden bei dem Vergleich im vorangegangenen Abschnitt außer Betracht gelassen, da sie bei jeder der angeführten Modellierungsarten benutzt werden können. Ihr Einsatz hilft im Normalfall, statische Berechnungen zu verbessern und diskrete Simulationen zu beschleunigen. Der Preis dafür ist gegebenenfalls eine geringere Exaktheit.

Kern dieser Arbeit ist die numerische Modellierung statistischer Kenngrößen von Transportsystem-Elementen (Kapitel 4). Zur Ermittlung dieser Kenngrößen ist eine Vielzahl von diskreten Simulationsläufen mit unterschiedlich parametrisierten Modellen einzelner Elemente nötig. Für jede der interessierenden Kenngrößen wird ein neuronales Netz erstellt, das sie für die gesamte Modellklasse approximiert. Alle damit erhaltenen Kenngrößen-Funktionen werden in einem Metamodell für eine Klasse von Transportsystem-Elementen benutzt. In einem weiteren Modellierungsschritt (Kapitel 5) können diese Metamodelle miteinander verbunden werden, um komplexe Systeme strukturtreu zu modellieren.

# 3 Neuronale Netze

Schon 1943 beschrieben McCulloch und Pitts die Struktur eines Berechnungsnetzes, das die grundlegenden Signaleigenschaften eines natürlichen Neuronennetzes ausnutzt [McC43]. 1949 entwarf Hebb die nach ihm benannte Trainingsstrategie [Heb49]. Am MIT baute Rosenblatt 1962 ein elektromechanisches(!) neuronales Netz auf, das bereits in der Lage war, auf einer 20 x 20 Punkte großen Bitmatrix Ziffern zu erkennen [Ros62]. Wissenschaftler, wie Minsky [Min85] und Kohonen [Koh72], prägten die Entwicklung auf dem Gebiet. Die in Abschnitt 3.2 vorgestellte Trainingsmethode (*Backpropagation*) wurde 1974 von Werbos erstmalig vorgestellt [Wer74].

Während einige numerische Methoden mit vertretbarem Aufwand „von Hand“ zu bewältigen sind, ist der Einsatz neuronaler Netze<sup>1</sup> an Berechnungs-Hardware (z.B. Computer) gebunden. Einen starken Schub bekam deren Entwicklung deshalb in den achtziger Jahren des vorigen Jahrhunderts durch die Verfügbarkeit preiswerter Rechner. Hopfield führte 1985 eine neue Netzstruktur ein und benutzte sie, um komplexe Optimierungsprobleme zu lösen [Hop85]. Das *Backpropagation*-Verfahren wurde noch einmal beschrieben, diesmal von Rumelhart, Hinton und Williams [Rum86]. Eine sehr lesenswerte Übersicht über die Entwicklung der neuronalen Netze am MIT findet man bei [Bro91].

Die Entwicklung künstlicher neuronaler Netze geht mit der Suche nach einer Methode für die implizite Repräsentation von Wissen einher. Beheimatet auf dem Gebiet der künstlichen Intelligenz, liegen die Schwerpunkte der Forschung auf der Erkennung von Schrift und Zeichnungen, von natürlichen Szenen und Sprache. Ihre Bedeutung für die Mustererkennung ist deshalb so hoch, weil für die Beschreibung von realen, also unscharfen Objekten mit Hilfe neuronaler Netze keine expliziten Regeln formuliert werden müssen; im Normalfall genügt es, die Netze mit einer genügend großen Menge von repräsentativen Beispielen zu „trainieren“ .

Auch für die Interpolation glatter Funktionen eignen sich künstliche neuronale Netze. Insbesondere auch dann, wenn die Dimension der numerischen Probleme hoch ist und wenn keine Singularität im betrachteten Definitionsbereich liegt, liefern sie gute Lösungen [Wan93]. Zwar verlangt die Trainingsphase viel Rechenzeit, aber die Berechnung der Netzantwort geht sehr schnell vonstatten.

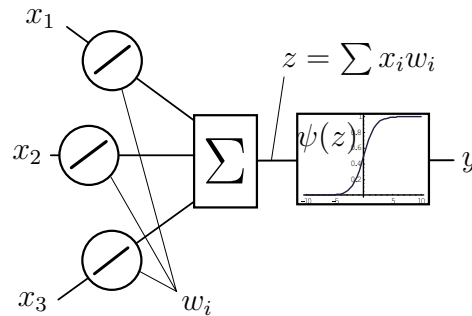
In diesem Kapitel wird das Konzept „Neuronales Netz“ erläutert. Ziel ist es, dem Leser zu vermitteln, dass die „künstliche Intelligenz“ hier lediglich durch einen relativ einfachen numerischen Ansatz realisiert wird.

---

<sup>1</sup>Da Verwechslungen mit natürlichen neuronalen Netzen und anderen netzartigen Berechnungsstrukturen in dieser Arbeit nicht drohen, wird statt der vollständigen Bezeichnung „künstliches neuronales Netz“ hier oft auch verkürzend „neuronales Netz“ oder nur „Netz“ benutzt.

### 3.1 Aufbau neuronaler Netze

Grundbaustein eines neuronalen Netzes ist ein abstraktes Neuron (Abbildung 3.1).



**Abbildung 3.1:** Struktur eines Neurons

Die Eingangswerte  $x_i$  ( $i = 1, 2, \dots$ ) werden gewichtet summiert<sup>2</sup>  $z = \sum_i w_i x_i$  und einer so genannten Aktivierungsfunktion  $\psi$  als Argument übergeben. Der Ausgangswert  $y$  eines Neurons berechnet sich dann mit

$$y = y(x_1, x_2, \dots) = \psi\left(\sum_i w_i x_i\right). \quad (3.1)$$

Die Funktion  $\psi : \mathbb{R} \rightarrow [0, 1]$  ist dabei monoton wachsend. In der Praxis werden vor allem

- stückweise lineare Funktionen,
- Schwellwertfunktionen, und
- sigmoide oder logistische Funktionen genutzt.

Wird das neuronale Netz für die Approximation von stetig differenzierbaren Funktionen benutzt, verwendet man für gewöhnlich die logistische Funktion (siehe Abbildung 3.2). An dieser Stelle werden deshalb kurz einige ihrer Eigenschaften dargestellt.

Die Funktion ist auf ganz  $\mathbb{R}$  für positive, reelle Parameter  $b$  und  $c$  folgendermaßen definiert (e ... Eulersche Konstante):

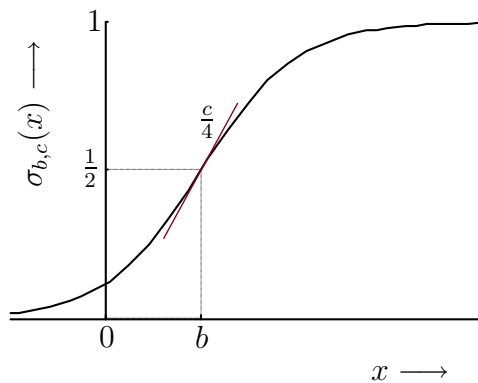
$$\sigma_{b,c}(x) = \frac{1}{1 + e^{c(b-x)}}. \quad (3.2)$$

Offensichtlich gilt  $0 < \sigma_{b,c}(x) < 1$ . Die Funktion ist streng monoton wachsend. Ihre Ableitung ist über ganz  $\mathbb{R}$  definiert:

$$\sigma'_{b,c}(x) = \frac{c e^{c(b+x)}}{(e^{bc} + e^{cx})^2}. \quad (3.3)$$

Die logistische Funktion hat genau einen Wendepunkt an der Stelle  $(b, 1/2)$ . Ihre Ableitung dort beträgt  $c/4$ . Das heißt, über die Parameter  $b$  und  $c$  können die Lage des Wendepunktes und der Anstieg an dieser Stelle festgelegt werden.

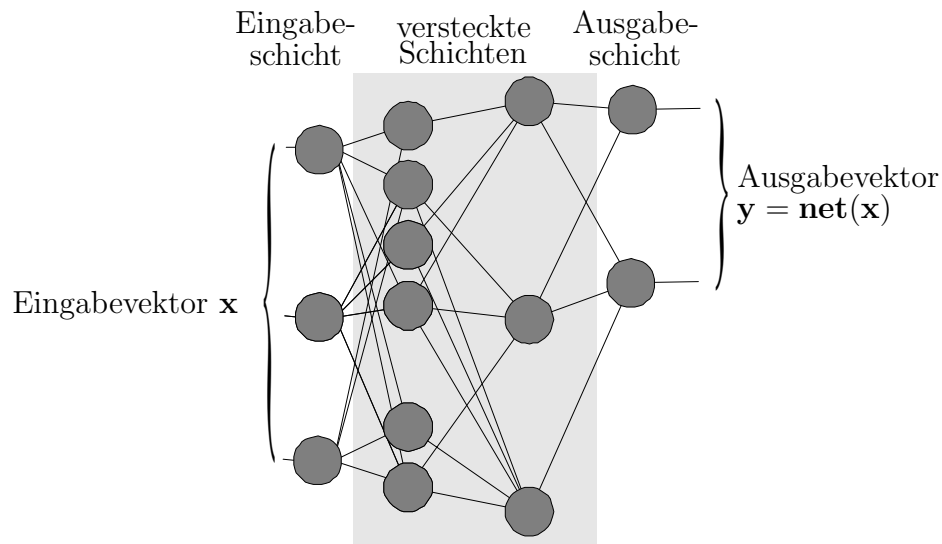
<sup>2</sup>Es werden hierbei alle Eingänge des Neurons berücksichtigt. Zugunsten einer verständlichen Lesbarkeit wird auf die explizite Angabe von Summationsgrenzen verzichtet. Für eine vollständige Darstellung sei auf [Roj93] verwiesen.



**Abbildung 3.2:** Logistische Funktion  $\sigma_{b,c}(x)$

Werden Neuronen in einem Netz verknüpft, kann der Ausgabewert  $y$  gleichzeitig Eingangswert weiterer Neuronen sein. Die Neuronen, deren Eingänge von außen in das Netz gegeben werden heißen *Eingabeneuronen*. Die *Ausgabeneuronen* liefern den Ergebnisvektor des Netzes (*Netzantwort*).

Oft benutzt man für numerische Approximationen Linarkombinationen von parametrisierten Ansatzfunktionen. Im Unterschied dazu trifft man bei neuronalen Netzen auf eine Verkettung der Funktionen. Dafür hat es sich bewährt, die Neuronen in Schichten (*layer*) anzuordnen (siehe Abbildung 3.3). Zwischen der Eingabe- und der Ausgabeschicht liegen die versteckten Schichten (*hidden layers*).



**Abbildung 3.3:** Beispiel eines neuronalen Netzes

Diese Art der Datenverarbeitung, bei der viele primitive Zellen miteinander verbunden sind, wird als *Konnektionismus* bezeichnet [Sch97]. Die Informationen werden dabei in den Verbindungen gespeichert. Ein Ergebnis erhält man durch die Berechnung der Ausgaben der einzelnen Zellen (*parallel distributed processing*).

Der Ausgabevektor des neuronalen Netzes ergibt sich durch die schichtweise Berechnung der Funktionswerte von den Eingangsneuronen bis zu den Ausgängen. Das heißt, beginnend mit dem Eingabevektor  $\mathbf{x}$  an den Neuronen der Eingabeschicht wer-

den die Ausgabewerte berechnet. Erst dann, wenn eine Schicht vollständig berechnet wurde, wird mit der nächsten fortgefahren. Schicht für Schicht werden so die Ausgabewerte der Neuronen bestimmt, bis die Ausgabeschicht den Ausgabevektor  $\mathbf{y}$  als Ergebnis der Berechnung ausgibt. Das neuronale Netz berechnet auf diese Weise zu einem Eingangsvektor  $\mathbf{x} = (x_1, x_2, \dots)$  den zugehörigen Ausgabevektor  $\mathbf{y} = \mathbf{net}(\mathbf{x})$ .

Es gibt auch neuronale Netz mit Rückkopplungen (rekurrente Netze). Diese sind im Rahmen dieser Arbeit jedoch nicht von Interesse, da keine dynamische Modellierung beabsichtigt wird. Deswegen sollen ausschließlich so genannte *feed-forward*-Netze betrachtet werden.

## 3.2 Training mit Backpropagation

Ausgangspunkt für das Training eines neuronalen Netzes sind Trainingsdaten. Das ist eine Menge  $M_t = \{(\mathbf{x}_i, \mathbf{d}_i) \mid i = 1, 2, \dots\}$  von Eingangsvektoren  $\mathbf{x}_t$  und den dazugehörigen bekannten Zielvektoren  $\mathbf{d}_t$ . Das Ziel des Netzwerkstrainings ist es, die Parameter des Netzes so zu verändern, dass die Ausgabevektoren  $\mathbf{y}_t = \mathbf{net}(\mathbf{x}_t)$  die Zielvektoren  $\mathbf{d}_t$  möglichst gut nähern. Als „Stellgrößen“ für die Approximation stehen die Gewichte der Verbindungen zwischen den Neuronen sowie ggf. skalare Parameter der Aktivierungsfunktionen zur Verfügung.

Die folgende Schreibweise wird an dieser Stelle festgelegt: Ein hochgestellter Index an einer Größe bezeichnet die Schicht auf die sich diese Größe bezieht. Tiefgestellte Indizes bezeichnen Nummern von Neuronen:

$y_j^u \dots$  Ausgang des  $j$ -ten Neurons in der  $u$ -ten Schicht

$x_{ij}^u \dots$   $i$ -ter Eingang des  $j$ -ten Neurons in der  $u$ -ten Schicht

$w_{ij}^u \dots$  Gewicht der Kante vom  $i$ -ten Neuron der Schicht  $u - 1$  zum  $j$ -ten Neuron in der  $u$ -ten Schicht

$z_j^u \dots$  gewichtete Summe im  $j$ -ten Neuron in der  $u$ -ten Schicht

$\psi_j^u(\cdot) \dots$  Aktivierungsfunktion des  $j$ -ten Neurons in der  $u$ -ten Schicht

Gegeben sei ein neuronales Netz mit  $U$  Schichten für einen festen Eingabevektor  $\mathbf{x}$  und einen Trainingsvektor  $\mathbf{d} = (d_1, d_2, \dots)$ . Die Berechnung der Netzantwort  $\mathbf{net}(\mathbf{x}) = \mathbf{y} = (y_1^U, y_2^U, \dots)$  sei beendet: Alle Neuronen haben demnach definierte Ein- und Ausgabewerte, die für weitere Berechnungen zur Verfügung stehen.

Als Fehlermaß bietet sich wegen ihrer stetigen Differenzierbarkeit die Euklidische Norm  $\|\cdot\|_2$  an:

$$\varepsilon = \|\mathbf{d} - \mathbf{y}\|_2 = \sum_j (d_j - y_j^U)^2. \quad (3.4)$$

### 3.2.1 Einzelner Trainingsschritt

Mit einem gedämpften Gradientenverfahren (Dämpfung bzw. Lernfaktor  $\eta$ ) werden die Gewichte  $w_{ij}^u$  der Neuronen der  $u$ -ten Schicht verändert um den Betrag

$$\Delta w_{ij}^u = -\eta \frac{\partial \varepsilon}{\partial w_{ij}^u}. \quad (3.5)$$

Kern des *Backpropagation*-Verfahrens ist die Berechnung dieser partiellen Ableitungen von „hinten“ nach „vorne“. Im Folgenden wird diese Methode in gleicher Weise beschrieben.

Die Betrachtungen beginnen mit der Ausgangsschicht  $U$ . Dann folgt aus (3.4) für jedes Neuron  $j$  der Ausgangsschicht sofort

$$\frac{\partial \varepsilon}{\partial y_j^U} = 2(y_j^U - d_j). \quad (3.6)$$

Diese Größen sind direkt berechenbar. Eine Schicht weiter in Richtung Eingang ergibt sich der Einfluss der gewichteten Summen auf den Fehler folgendermaßen:

$$\frac{\partial \varepsilon}{\partial z_j^U} = \frac{\partial \varepsilon}{\partial y_j^U} \cdot \frac{\partial y_j^U}{\partial z_j^U}. \quad (3.7)$$

Der erste Faktor ist wegen (3.6) bekannt und den zweiten erhält man aus

$$\frac{\partial y_j^u}{\partial z_j^u} = \psi_j^{u'}(z_j^u) \quad (\text{für } 1 \leq u \leq U). \quad (3.8)$$

Im Normalfall liegt die Ableitung der Aktivierungsfunktion in einer sofort berechenbaren Form vor. Für die logistische Funktion ist sie mit (3.3) gegeben. Somit kann also auch Gleichung (3.8) berechnet werden.

Für die Gewichte der Verbindungen ergibt sich dann

$$\frac{\partial \varepsilon}{\partial w_{ij}^U} = \frac{\partial \varepsilon}{\partial z_j^U} \cdot \frac{\partial z_j^U}{\partial w_{ij}^U}. \quad (3.9)$$

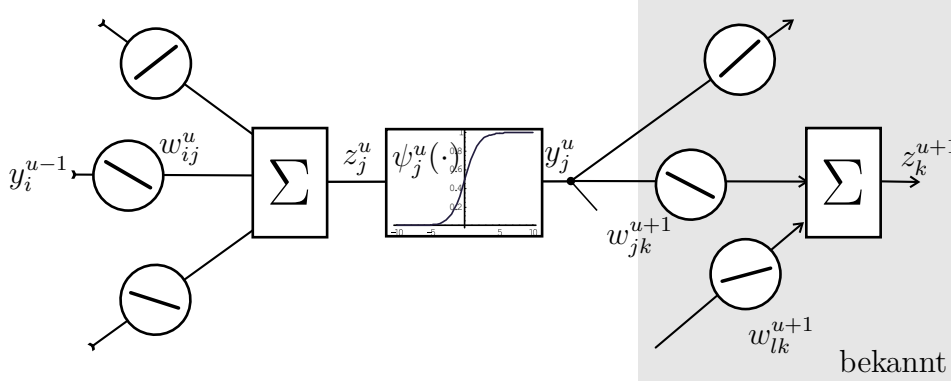
Und mit (3.7) ist mit

$$z_j^u = \sum_l w_{lj}^u y_l^{u-1} \quad (3.10)$$

wegen

$$\frac{\partial z_j^u}{\partial w_{ij}^u} = \frac{\partial}{\partial w_{ij}^u} \sum_l w_{lj}^u y_l^{u-1} = y_i^{u-1} \quad (\text{für } 1 < u \leq U) \quad (3.11)$$

auch (3.9) sofort anzugeben. Als Induktionsanfang sind somit die interessierenden Ableitungen für die Ausgangsschicht  $U$  bereitgestellt.



**Abbildung 3.4:** Neuron  $j$  in der Schicht  $1 < u < U$

Abbildung 3.4 hilft, den Induktionsschritt zu veranschaulichen. Vorausgesetzt sei dafür, dass alle interessierenden Ableitungen für die Schichten größer  $u$  berechnet sind.

Für die Summanden in

$$\frac{\partial \varepsilon}{\partial y_j^u} = \sum_k \frac{\partial \varepsilon}{\partial z_k^{u+1}} \cdot \frac{\partial z_k^{u+1}}{\partial y_j^u} \quad (3.12)$$

sind jeweils die ersten Faktoren bekannt. Offenbar ist

$$z_k^{u+1} = \sum_l y_l^u w_{lk}^{u+1}, \quad (3.13)$$

und somit erhält man für die zweiten Faktoren in (3.12)

$$\frac{\partial z_k^{u+1}}{\partial y_j^u} = w_{jk}^{u+1}. \quad (3.14)$$

Nachdem  $\frac{\partial \varepsilon}{\partial y_j^u}$  berechnet werden kann, erhält man mit (3.8)

$$\frac{\partial \varepsilon}{\partial z_j^u} = \frac{\partial \varepsilon}{\partial y_j^u} \cdot \frac{\partial y_j^u}{\partial z_j^u}. \quad (3.15)$$

Schließlich wird (3.11) benutzt, um mit

$$\frac{\partial \varepsilon}{\partial w_{ij}^u} = \frac{\partial z_j^u}{\partial w_{ij}^u} \cdot \frac{\partial \varepsilon}{\partial z_j^u} = y_i^{u-1} \frac{\partial \varepsilon}{\partial z_j^u} \quad (3.16)$$

den Induktionsschritt zu vollenden.

Diese Induktion wird bis zu den Eingabeneuronen schichtweise benutzt, um die partiellen Ableitungen des Fehlers nach allen Gewichten zu berechnen. Mit dem gedämpften Gradientenschritt

$$\Delta w_{ij}^u = -\eta \frac{\partial \varepsilon}{\partial w_{ij}^u}$$



können nun die Gewichte verändert werden:

$$w_{ij}^u := w_{ij}^u + \Delta w_{ij}^u. \quad (3.17)$$

Die Wahl des Dämpfungsfaktors  $\eta$  ist entscheidend für das Konvergenzverhalten des Verfahrens. Insbesondere höhere Dimensionen der Eingangsvektoren legen nahe, größere Dämpfungen zu benutzen. Bei den in Kapitel 4 betrachteten Fällen wurde für den Start des Trainings  $\eta = 0.5$  gewählt, damit begeben sich die Gewichte schnell in den Erfolg versprechenden Bereich. Schritt für Schritt wurde die Dämpfung dann aber bis auf  $\eta = 0.01$  verringert, um das lokale Minimum des Netzfehlers gut zu finden.

Auf ähnliche Weise können die Parameter der Aktivierungsfunktionen angepasst werden: Sei  $\alpha$  ein Parameter der Funktion  $\psi_j^u$ . Dann ist

$$\frac{\partial \varepsilon}{\partial \alpha} = \frac{\partial \varepsilon}{\partial y_j^u} \cdot \frac{\partial y_j^u}{\partial \alpha} \quad (3.18)$$

Wegen (3.12) benötigt man nur noch die partiellen Ableitungen der benutzten Aktivierungsfunktion nach ihren Parametern, um mit

$$\alpha := \alpha - \eta \frac{\partial \varepsilon}{\partial \alpha} \quad (3.19)$$

den jeweiligen Parameter entsprechend anzupassen. Für die sigmoide Aktivierungsfunktion  $\sigma_{b,c}(x)$  ergibt sich beispielsweise

$$\frac{\partial \sigma_{b,c}(x)}{\partial b} = -\frac{c e^{c(b+x)}}{(e^{bc} + e^{cx})^2} \quad \text{bzw.} \quad \frac{\partial \sigma_{b,c}(x)}{\partial c} = \frac{e^{c(-b+x)} (-b+x)}{(1 + e^{c(-b+x)})^2}. \quad (3.20)$$

Die Initialisierung der Gewichte und Parameter des neuronalen Netzes wird gewöhnlich mit zufälligen Startwerten zwischen -1 und 1 durchgeführt [Bab97].

### 3.2.2 Trainingszyklen

Durch wiederholte Anwendung des Trainingsschrittes wäre es möglich, die Parameter des neuronalen Netzes so zu bestimmen, dass der resultierende Fehler  $\|\mathbf{net}(\mathbf{x}) - \mathbf{d}\|_2$  beliebig klein wird. Allerdings soll das Netz nicht nur für *einen* Ein- bzw. Ausgabevektor trainiert werden. Es soll *alle* Vektoren  $(\mathbf{x}, \mathbf{d}) \in M_t$  approximieren.

Aus diesem Grund wird das Gradientenverfahren bereits nach einem oder wenigen Schritten abgebrochen und mit dem nächsten Trainingsdatensatz  $(\mathbf{x}, \mathbf{d}) \in M_t$  fortgeführt. Ein *Trainingszyklus* umfasst die Durchführung von Gradientenschritten für *alle* Trainingsdaten  $M_t$ .

Um alle Daten gleichermaßen zu berücksichtigen, empfiehlt es sich, für jeden Zyklus die Reihenfolge des Trainings der einzelnen Datensätze zufällig zu wählen.

Für das Training der in Kapitel 4 benutzten Netze wurden zwischen 10 000 und 100 000 Zyklen benutzt. Während zu Beginn des Trainings eine Beobachtung empfohlen wird, um sich von der Konvergenz der Netzfehler zu überzeugen, kann später darauf verzichtet werden.

### 3.3 Einsatz neuronaler Netze

Das klassische Einsatzgebiet neuronaler Netze, die Klassifikation von Mustern, soll hier nicht berücksichtigt werden. Stattdessen geht es zunächst um Arbeiten, bei denen neuronale Netze für die Modellierung von Transportsystemen oder in verwandten Gebieten benutzt werden.

In der Produktionsplanung werden neuronale Netze im Wesentlichen zur Vorhersage von Bedienzeiten und Fertigungszeiten der Produkte benutzt.

Schmidt [Sch99] benutzt künstliche neuronale Netze, um ein schnelles Prognosewerkzeug für die Produktionsplanung in der *Front-End*-Fertigung der Halbleiterherstellung zu schaffen. Schwankungen der Durchlaufzeiten der Produkte werden in die Prognose einbezogen. Weber [Web99] benutzt neuronale Netze zur Vorhersage der Fertigungszeit bei stark variierender Produktpalette.

Ostermann benutzt ein hybrides Modell, um die Durchlaufzeiten bei der Automobilproduktion zu modellieren [Ost01]. Dabei werden ganze Fertigungsbereiche mit Hilfe von neuronalen Netzen erfolgreich abgebildet.

Sabuncuoglu und Touhami untersuchen die Stabilität beim Einsatz neuronaler Netze als Metamodelle zur Prognose von Produktionsdaten [Sab02]. Damit konnten in Echtzeit optimale *dispatching*-Regeln des Transportsystems ausgewählt werden.

Für ein einfaches Fertigungssystem demonstrieren Nasreddin und Mollaghasemi, dass ihre eingesetzten neuronalen Netze bessere Ergebnisse für die Modellierung der Produktionszeiten liefern als statistische Regressionsmodelle [Nas99].

Schulze benutzt rekurrente neuronale Netze, um die voraussichtliche Verweilzeit in einem System mit einer Bedienstation zu prognostizieren [Sch00b].

All diese Arbeiten belegen, dass es möglich ist, Bedienstationen oder komplette Fertigungsbereiche mit Hilfe von neuronalen Netzen abzubilden. Im Unterschied zu den hier betrachteten Fällen sind Transportsystem-Elemente jedoch instand, ankommende Transporteinheiten entsprechend gewisser Regeln zu verteilen. Aus diesem Grunde sollen nun aktuelle verkehrswissenschaftliche Arbeiten diskutiert werden.

Chang und Su benutzen neuronale Netze, um Staus im Straßenverkehr vorherzusagen [Cha95]. Dabei geht es um Bewegungen von Fahrzeugen auf einer mehrspurigen Straße zwischen zwei Kreuzungen. Ihnen gelingt es, ausgehend von der aktuellen Belastungssituation kurzfristige Prognosen zu liefern. Sie benutzen dafür ein Netz mit nur einer versteckten Schicht. In ihrem Beitrag bedauern sie mehrfach, dass der einzige Weg für den Entwurf einer optimalen Netztopologie eine mehr oder weniger systematische Heuristik ist.

Weitere Arbeiten aus dem Bereich des Verkehrswesens: Yang und Qiao [Qia98] benutzen neuronale Netze um die Belastung von Autobahnen zu klassifizieren. Die Eingabemuster für die neuronalen Netze sind dabei nicht Sprache oder Bilder sondern Geschwindigkeiten von Fahrzeugen. In einer weiteren Arbeit [Qia01] benutzen sie neuronale Netze, um das Verhalten von Fahrzeugen hinter einer Ampel zu modellieren. Unter Berufung auf Cybenko [Cyb89] benutzen sie hierzu ein neuronales Netz mit nur einer versteckten Schicht.

Obwohl diese verkehrswissenschaftlichen Arbeiten sich mit der Modellierung und Vorhersage von Staus beschäftigen, sind die Ergebnisse nicht direkt auf innerbetriebliche Transportsysteme übertragbar. Der wesentliche Grund dafür ist, dass Überholvorgänge oder objektspezifische („individuelle“) Bewegungen der Transporteinheiten in den hier betrachteten automatischen Transportsystemen nicht abgebildet werden.

Auch für die Bewertung und Regelung von Datenströmen in Kommunikationsnetzen werden neuronale Netze erfolgreich genutzt [Rau88], doch hier sind wesentlich andere Grundvoraussetzungen als für Gutströme in Transportsystemen gültig. Insbesondere sind Materialfluss-Systeme selten so komplex, dass für einen Transportauftrag eine Vielzahl von unterschiedlichen Wegen benutzt werden kann.

Diese Arbeiten, obwohl nicht direkt auf die hier interessierende Problematik übertragbar, liefern jedoch wertvolle Hinweise auf die Machbarkeit und den Sinn derartiger Modellierungen überlagerter Ströme. Abschließend werden hier Arbeiten über die Approximation von (mehrdimensionalen) Funktionen mit neuronalen Netzen betrachtet. Hierzu existieren Aussagen zu Topologie, Konvergenz und Konvergenzgeschwindigkeit.

Chiu u.a. untersuchen die Stabilität der Berechnungen von neuronalen Netzen [Chi96]. Im Vergleich mit einer optimalen (minimalen) Topologie erweist sich eine Struktur mit redundanten Neuronen als robuster gegenüber leicht veränderten Eingaben.

Cybenko hat nachgewiesen, dass eine reellwertige Funktion mit beschränktem Anstieg beliebig genau approximiert werden kann mit einem neuronalen Netz mit nur einer versteckten Schicht [Cyb89]. Dabei benutzt er sigmoide Funktionen als Aktivierungsfunktionen. Die Anzahl der Neuronen in dieser Schicht kann allerdings sehr groß werden. In Abhängigkeit vom Fehler der Rohdaten können für Approximation gewisser Funktionen in [Bur00] untere Grenzen für die Anzahl der Neuronen für eine gewünschte Güte der Approximation gegeben werden. Betrachtungen zur Stabilität solcher Netze finden sich bei [Lo01]

Kratzer betont, dass die Struktur der Netze insbesondere für die Konvergenzgeschwindigkeit eine wesentliche Rolle spielt [Kra90]. Die einzige Möglichkeit, die Struktur neuronaler Netze zu optimieren, sieht Moody in gezielter heuristischer Suche [Moo94]. Abschätzungen über die Güte von Approximationen finden sich in [Mha97], [Mai00] und [Mha98]. Der Rechenaufwand (Komplexität) für das Training von neuronalen Netzen mit  $n$  Neuronen in einer versteckten Schicht wird bei Kůrkova und Sanguineti mit  $O(\sqrt{n})$  für sigmoide Aktivierungsfunktionen berechnet [Kur01].

Wavelets sind orthogonale Basisfunktionen des reellwertigen Funktionenraums (eine Einführung etwa bei [Gra95]). Sie werden zum Beispiel bei der Signalverarbeitung (Fourier-Analyse) eingesetzt. Rao und Pappu [Rao96] benutzen sie mit Erfolg als Aktivierungsfunktionen in neuronalen Netzen. Mhaskar [Mha96] benutzt ein neuronales Netz, das eine Funktion punktweise approximiert, um dann die Funktion insgesamt mit Hilfe von Wavelets zu nähern.

Im folgenden Kapitel wird ein Modell vorgestellt, das für einige Klassen wichtiger Transportsystem-Elemente statistische Kenngrößen numerisch approximiert. Dafür erwiesen sich neuronale Netze als gut geeignet.

Ein neuer Modellierungsansatz erlaubt es darüber hinaus, diese einzelnen Teilmodelle zu einem größeren System zusammenzusetzen (Kapitel 5).

# 4 Modell für Elemente von Materialfluss-Systemen

Zu den gegenwärtig vorhandenen Möglichkeiten, Materialfluss-Systeme zu modellieren (siehe Abschnitt 2.3) wird nun eine weitere hinzugefügt. Der Grundgedanke des neuen Modells ist es, bestimmte statistische Kenngrößen eines Materialflusselements mit Hilfe neuronaler Netze zu modellieren.

Der erste Abschnitt dieses Kapitels erläutert detailliert die einzelnen Schritte der Modellbildung. Dabei werden an jeder Stelle die jeweiligen modifizierten Modellvoraussetzungen verdeutlicht. Das auf diese Weise hergeleitete parametrische Modell wird in einer letzten Stufe durch neuronale Netze numerisch approximiert. Das ist jedoch nicht mehr im Rahmen von allgemeinen Betrachtungen möglich.

Die folgenden Abschnitte widmen sich deshalb konkreten Modellen einiger grundlegender Transportsystem-Elemente. Mit Hilfe umfangreicher Daten aus Simulationen konnten neuronale Netze so trainiert werden, dass sie die jeweiligen statistischen Kenngrößen der Transportsystem-Elemente gut abbilden.

Detailliert modelliert werden die folgenden Elemente:

- Einschleusung mit absoluter Vorfahrt,
- Ausschleusung von einer Haupt- auf eine Nebenstrecke,
- Kreuzung mit 2 Ein- und Ausgängen (z.B. Hub- oder Drehtisch),
- Regalbediengerät,
- Bedienstation mit gammav verteilter Bedienzeit.

In den entsprechenden Abschnitten werden die elementspezifischen Parameter diskutiert und die Ergebnisse des Trainings der neuronalen Netze ausgewertet.

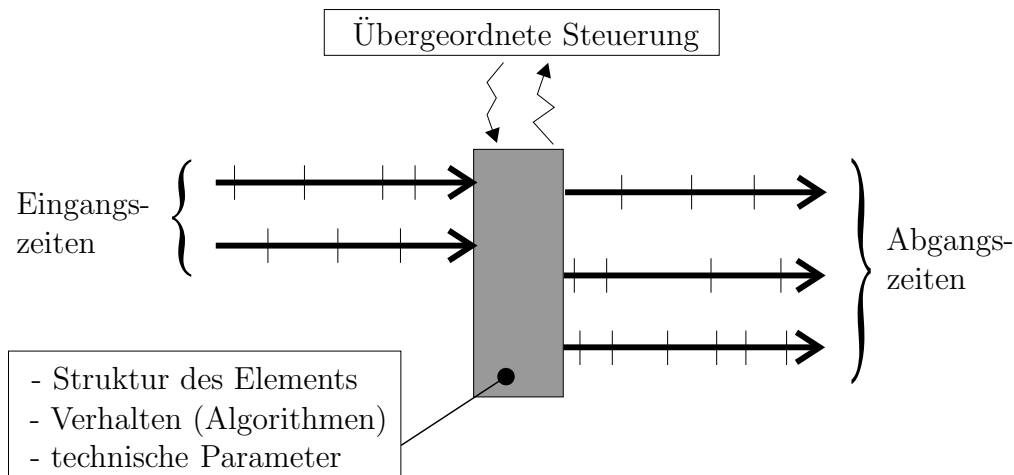
Im Rahmen dieser Arbeit werden lineare Förderstrecken anders als die obigen Elemente abgebildet. Unter der Annahme, dass Staus nicht durch diese Förderer verursacht werden, interessieren hier nur ihre Fähigkeiten, Transporteinheiten von einem Punkt zu einem anderen zu befördern und Stauplätze vor anderen Elementen bereitzustellen.

## 4.1 Modellbildung

Grundlage dieses Modells ist die Betrachtung diskreter Ereignisse (S. 18f.). Es interessiert nicht primär, welche Prozesse eine Transporteinheit in einem Transportsystem-Element durchläuft sondern, *wann* dieses Element betreten und wann es wieder verlassen oder freigegeben wird. Es ist hierbei unerheblich, wie groß oder wie schnell ein Element ist, oder auf welche Weise es seine Aufgabe erfüllt. Wichtig ist einzig, wieviel Zeit ein Element für die Bearbeitung eines Transportauftrages benötigt und wann die nächste Transporteinheit abgefertigt werden kann. Diese Herangehensweise entspricht den Modellannahmen der diskreten Simulation (Abschnitt 2.3.4).

Räumliche Beziehungen zwischen Transportsystem-Elementen spielen nur insofern eine Rolle, als dass sie die logistische Struktur des Systems widerspiegeln<sup>1</sup>.

Der allgemeinen Fall eines Elementes in einem Materialfluss-System kann folgendermaßen beschrieben werden (siehe Abbildung 4.1):



**Abbildung 4.1:** Allgemeines ereignisdiskretes Transportsystem-Element

Über verschiedene Eingänge treffen Forderungen ein.<sup>2</sup> Die Abstände zwischen den einzelnen Eingangszeiten können beliebig sein. Dass Transporteinheiten in der Realität, schon allein wegen ihrer physischen Ausdehnung, nicht mit *beliebig* kurzen Zwischenzeiten eintreffen können, wird für die Modellierung später berücksichtigt, soll aber an dieser Stelle noch außer Acht gelassen werden.

Entsprechend seiner Aufgabe im System reagiert das betrachtete Element auf eintreffende Forderungen. Dazu werden in der Realität bestimmte Steuerungs- und Transportprozesse ausgeführt, die in der ersten Modellierungsstufe (diskrete Simulation) mit Hilfe von Strukturen und Algorithmen abgebildet werden können. Steuerungsregeln und physikalisch-technische Prozesse benutzen einen Satz von element-spezifischen Parametern, wie etwa Übergabezeiten, Transfargeschwindigkeiten und

<sup>1</sup>Auf Lagebeziehungen, wie die Unterscheidung zwischen Einschleusungen „von rechts“ oder „von links“ kann aus Symmetriegründen verzichtet werden.

<sup>2</sup>Das müssen nicht ausschließlich reale Güter sein, auch „Datenpakete“ (wie z.B. Auslagerungsaufträge) sind hier zugelassen.

dergleichen. Unter gewissen Umständen wird eine übergeordnete Steuerung kontaktiert, um beispielsweise Prioritäten bei der Abarbeitung von mehreren Aufträgen zu ermitteln. Zu einem *durch das Element* definierten Zeitpunkt verlässt die Transporteinheit das Materialfluss-Element über einen der Ausgänge.

Wenn  $n_E$  und  $n_A$  die Anzahlen der Ein- bzw. Ausgänge des Transportsystem-Elements sind, werden die Ankunftszeiten des  $i$ -ten Eingangs mit  $\tau_{E_i}$  ( $i = 1, 2, \dots, n_E$ ) und die Abgangszeiten mit  $\tau_{A_j}$  (für  $j = 1, 2, \dots, n_A$ ) bezeichnet. Beginnend mit dem Start der Beobachtung (z.B. zu einer Systemzeit  $\tau = 0$ ), ergeben sich somit Folgen von Zeiten  $(\tau_{E_i})_k$  ( $k = 1, 2, \dots$ ) und  $(\tau_{A_j})_k$  ( $k = 1, 2, \dots$ ).

Obwohl die Abläufe in einem Transportsystem weitgehend deterministisch sind, spielt auch der Zufall eine Rolle. Damit sind nicht nur zufällige Abweichungen der realen von den modellierten Zeiten, etwa für eine Lastübernahme, gemeint. Forderungen von außen, wie beispielsweise Transportaufträge aus der Produktion oder der Kommissionierung, haben in weit größerem Umfang einen stochastischen Einfluss. Aus der „Sicht“ eines einzelnen Transportsystem-Elements können alle Ankünfte von Forderungen als zufällig betrachtet werden - egal, ob sie, in einem größeren Zusammenhang betrachtet, deterministisch oder wirklich stochastisch sind.

#### 4.1.1 Lokalitätsbedingungen

Für die Modellbildung ist es wesentlich, dass keine Transportsystem-Element vom Zustand eines anderen Elements beeinflusst wird. Das führt zu den folgenden zwei Lokalitätsbedingungen:

##### Übergeordnete Steuerung

Die Folgen  $(\tau_{E_i})_k$  und  $(\tau_{A_j})_k$  können als *stochastische Prozesse* [Fis76] betrachtet werden. Offenbar korrelieren die Aus- und Eingangsprozesse. Auch wenn das Modell des Transportsystem-Elements selbst keine zufälligen Größen beinhaltet, ist noch keine direkte funktionelle Abhängigkeit der Eingangs- von den Abgangsprozessen gegeben, da eine übergeordnete Steuerung (etwa zur Stauvermeidung) direkt in die Steuerung des Elements eingreifen kann. Möchte man solche Regelmechanismen nachbilden, ist man auf eine realitätsnahe Simulation angewiesen. Für die hier vorgestellte Modellierung wird angenommen, dass das Verhalten des Transportsystem-Elements vollständig selbstbestimmt ist. Eine externe Steuerung wird ausgeschlossen.

Falls eine externe Steuerung im Wesentlichen nur für die Routenplanung benötigt wird, ist eine Modellierung durch die Verwendung von lokalen Transportmatrizen möglich: für jedes Transportsystem-Element wird damit statistisch festgelegt, wie die ankommenden Transporteinheiten auf die Ausgänge des Elements zu verteilen sind.

## Blockierungen

Während sich andere Arbeiten (etwa [Ral98] oder [Fur92]) intensiv mit Blockieretzwerken beschäftigen, soll hier ein anderer Weg eingeschlagen werden. Obwohl bei einem gut dimensionierten Transportsystem Blockierungen einzelner Elemente nicht in jedem Falle kritisch sein müssen, bietet die Kenntnis der sich entwickelnden Warteschlangen eines nicht blockierten Systems zumindest wichtige Anhaltspunkte für die Auslegung der Pufferplätze. In Ergänzung zur oben formulierten Forderung werden in den hier betrachteten Modellen Blockierungen ausgeschlossen. Das bedeutet, kein Transportsystem-Element wird bei der Abarbeitung seiner Forderungen durch ein anderes Element behindert.

Transportsysteme, bei denen Blockierungen oder Betriebseinschränkungen einzelner Elemente als Normalfälle angesehen werden, können mit diesem Modell nur bedingt abgebildet werden.

Diese beiden Lokalisierungsbedingungen (Ausschluss einer übergeordneten Steuerung und Verbot von Blockierungen) schränken den Modellbereich auf dem ersten Blick stark ein. Für ein Transportsystem, das im Mittel der Zeit gut ausgelastet ist, kommt es in Spitzenzeiten unter Umständen zu Blockierungen einzelner Elemente. Mit Hilfe globaler Strategien zur Stauvermeidung können dabei jedoch unerwünschte Nebeneffekte oft verhindert werden.

Mit dem vorgestellten Modell ist aber auf der anderen Seite eine quantitative Bewertung der Belastung spezieller Transportsystem-Elemente bereits frühzeitig möglich. Aussagen über die Anzahl benötigter Stauplätze für eine gewisse statistische Sicherheit dienen bei der Grobplanung als gute Kenngrößen zur Dimensionierung des Stauraumes.

### 4.1.2 Egalisierung

Die Steuerung eines Transportsystem-Elements benutzt in der Regel Identifikatoren an den Transporteinheiten, um zu entscheiden, wie sie zu handhaben sind. Das betrifft insbesondere die Entscheidung, zu welchem Ausgang des Elements eine ankommende Transporteinheit befördert wird. Aber auch Bearbeitungszeiten in Maschinen sind im Allgemeinen vom konkreten Werkstück abhängig.

Ab dieser Modellierungsstufe sind zwei Transporteinheiten nicht mehr unterscheidbar.

Durch die Abkehr von einer globalen Steuerung wird die Kommunikation zwischen verschiedenen Transportsystem-Elementen unterbunden: Damit ist keine Koordination zwischen ihnen mehr möglich. Deshalb erscheint dieser Schritt der Egalisierung als nicht drastisch. Sollen Systeme, bei denen die Abfertigung an den einzelnen Transportsystem-Elementen im Los- oder *Batch*-Betrieb erfolgt, exakt betrachtet werden, ist dieser Modellierungsschritt unter Umständen nicht mehr akzeptabel, eine diskrete Simulation erscheint dann unabdingbar.

Für fördertechnische Elemente werden ab dieser Modellierungsstufe lediglich die Häufigkeiten der vorkommenden Eingangs-Ausgangs-Bewegungen (lokale Transportmatrix) innerhalb des Elements berücksichtigt. Zufällig, aber entsprechend den rea-



len Häufigkeiten, werden die ankommenden Transporteinheiten zu den jeweiligen Ausgängen befördert. Bearbeitungsstationen, die für verschiedene Produkte unterschiedliche Zeiten benötigen, werden analog abgebildet. Umrüstzeiten können dabei nicht gesondert betrachtet werden. Auch hier gilt der Verweis auf die diskrete Simulation im Bedarfsfall.

### 4.1.3 Statistische Modellierung der Ströme

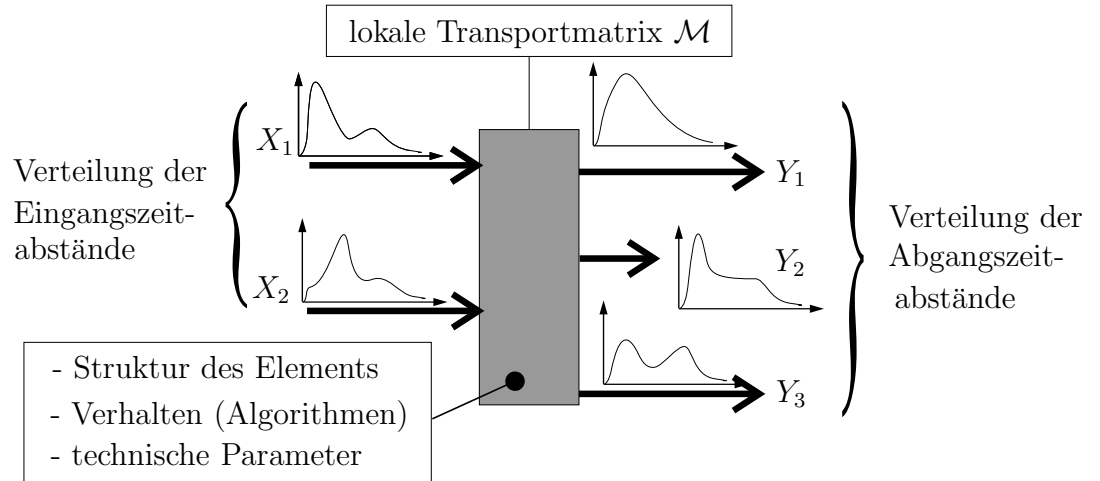
Mit dem nächsten Modellierungsschritt geht man zu einer stationären, statistischen Sichtweise über (siehe Abbildung 4.2).

Da Transporteinheiten (d.h. Ereignisse) eines Stroms nicht mehr voneinander unterscheidbar sind, kann die Schreibweise vereinfacht werden.  $(\tau)_k$  sei ein Eingangs- oder Abgangsprozess des betrachteten Elements. Mit  $t_k = \tau_{k+1} - \tau_k$  erhält man die entsprechenden *Zwischenzeiten*.

Und mit

$$F(t) = P(t_k < t \mid k \in \mathbb{N}) \quad (4.1)$$

ergibt sich die zugehörige statistische Verteilungsfunktion.



**Abbildung 4.2:** Stationäres, statistisches Modell eines Transportsystem-Elements

Die Zufallsgrößen  $X_i$  und  $Y_j$  stehen für die zufälligen Zwischenzeiten an den Eingängen  $i = 1, 2, \dots, n_E$  bzw. den Ausgängen  $j = 1, 2, \dots, n_A$ .

Als **Eingangsverteilungen** seien zur Verkürzung der Schreibweise die statistischen Verteilungen  $F_{X_i}$  der Differenzen der Ankunftszeiten an den Eingängen  $i = 1, 2, \dots, n_E$  bezeichnet. **Abgangsverteilungen** seien analog die statistischen Verteilungen  $F_{Y_j}$  der Abstände der Abgangsszeiten an den Ausgängen ( $j = 1, \dots, n_A$ ).

Mit diesem Modellierungsschritt gehen insbesondere zeitliche Verläufe verloren. Einschwingprozesse, Schwankungen infolge von Schichtbetrieb und dergleichen können ab dieser Modellierungsstufe nicht mehr modelliert werden.

Der Grund dafür soll hier kurz beispielhaft veranschaulicht werden. Angenommen, es gibt zwei Betriebsphasen des Systems, etwa Nacht- und Tagschichten. In der Nacht wird weniger produziert, die Belastung des Transportsystems ist geringer als

am Tage. Das spiegelt sich in den Prozessen der Eingangs- und Abgangszeiten exakt wider: Es gibt lange Phasen mit geringen Ankunftszeitabständen und Phasen mit deutlich größeren Abständen. In den Verteilungsfunktionen der Zeitabstände erkennt man, dass die Abstände in etwa 67 Prozent der Fälle kurz sind, und für die restlichen 33 Prozent deutlich höhere Abstände auftreten. Nicht ersichtlich ist dort jedoch, *wann* diese verschiedenen Belastungen auftreten. Für ein Transportsystem-Element könnte man also Eingangsprozesse finden, deren Verteilungsfunktionen zwar diesem Schichtbetrieb entsprechen, die aber statt dessen für einen konstanten Betrieb des Systems stehen.

Als Lösung dieses Problems könnte die interessierende Zeit in quasistationäre Abschnitte unterteilt werden, die dann einzeln modelliert werden. Insbesondere bei langfristigen Schwankungen (etwa jahreszeitlichen oder wöchentlichen) ist dieses Vorgehen sinnvoll. Sind die Abschnitte deutlich kürzer, muss man bei der Modellierung in Rechnung stellen, dass beispielsweise Warteschlangen eine gewisse Zeit brauchen, um sich aufzubauen. Daten, die aus statischen Modellläufen von bis zu 50 Tagen Länge gewonnen werden, sind nicht ohne Weiteres auf Systeme in ihrer Einschwingphase übertragbar.

**Bemerkung:** Diskrete Simulationen werden hier dazu benutzt, um statistische Kenngrößen von Teilsystemen zu ermitteln. Das bedeutet, die Simulation wird in der Regel beendet, wenn ausreichend Daten zur genauen Abschätzung der Kenngrößen erzeugt wurden. Das Simulationsende ist im Normalfall dann erreicht, wenn im Verlauf der Simulation der Betrag der Änderung gewisser statistischer Schätzwerte des Elements (etwa für Quantile von Warteschlangenlängen oder für Variationskoeffizienten) vorgegebene Schranken unterschreitet.

Die Länge der Simulationsläufe für ein gegebenes Element hängt in erster Linie von der Anzahl der eintreffenden Forderungen ab. Sie wird jedoch auch deutlich von der lokalen Transportmatrix und anderen Parametern beeinflusst und unterscheidet sich von Element zu Element. Da das Simulationsende über ein statistisches Abbruchkriterium dynamisch bestimmt wird, ist die Anzahl der im Verlauf der Simulation verarbeiteten Ereignisse nebensächlich. Sie wurde bei den durchgeführten Simulationen nicht protokolliert.

Die wiederholt in dieser Arbeit angegebenen Simulationslängen sind Schätzungen, die auf den Erfahrungen bei der Arbeit mit den Modellen beruhen. Sie sollen lediglich einen Eindruck über die Größenordnung der in der Simulation modellierten Zeitdauer vermitteln.

Dem Begriff „Simulationstag“ in der Modellwelt steht die „Rechenzeit“ gegenüber. Auch diese tatsächlich benötigte Zeit, um die Kenngrößen des Modells zu ermitteln bzw. zu approximieren, spielt eine Rolle für die Festlegung der Abbruchkriterien: Bei 4000 Simulationsläufen fällt ein Unterschied zwischen Rechenzeiten von 5 oder 50 s recht deutlich ins Gewicht. Unabhängig von der zur Verfügung stehenden Berechnungshardware ist deshalb eine Balance zwischen der Genauigkeit der statistischen Parameter und der Rechenzeit zu finden.

In diesem Modellierungsschritt wurden die Eingangs- und Ausgangsprozesse mit Hilfe ihrer Verteilungsfunktionen abgebildet. Unabhängig davon bleiben die Bedienprozesse des Transportsystem-Elements unverändert. Die lokale Steuerung wird dabei nicht beeinflusst.

#### 4.1.4 Parametrisches Modell

Bis zu dieser Stelle ist in diesem Modell ein Transportsystem-Element durch seine innere Struktur, Steuerung und Parameter abgebildet. Diese Sammlung von Eigenschaften soll nun strukturiert werden, um die am Ende des Modellbildungsprozesses stehenden numerischen Approximation zu ermöglichen.

Dazu werden zunächst Struktur und Steuerungsalgorithmen des Transportsystem-Elements zusammengefasst als **Klasse des Elements**. Zwei Transportsystem-Elemente gehören derselben Klasse an, wenn sie

- a) die gleiche Struktur besitzen und
- b) die gleichen Steuerungsregeln benutzen.

Verschiedene Repräsentanten einer Klasse unterscheiden sich lediglich durch die klassenspezifischen Parametertupel  $\pi_K$ . Dazu gehören Lastübergabe- und Positionierzeiten. Aber auch geometrische Größen können durch numerische Parameter repräsentiert werden. Die für konkrete Klassen benutzten Parameter werden in den nachfolgenden Abschnitten ausführlich erläutert. Die Dimension des Parametertupels  $\pi_K$  kann von Klasse zu Klasse verschieden sein.

Die Einteilung in Klassen bringt keine Einschränkungen mit sich, sie ist ein rein pragmatischer Schritt, um die nachfolgende numerische Modellierung praktisch zu ermöglichen.

Offenbar sind ein Regalbediengerät (RBG) und eine Verzweigung wegen ihrer strukturellen Unterschiede verschiedenen Klassen zuzuordnen. Doch auch zwei RBG können unterschiedlichen Klassen angehören, wenn sie unterschiedliches Parkverhalten<sup>3</sup> haben.

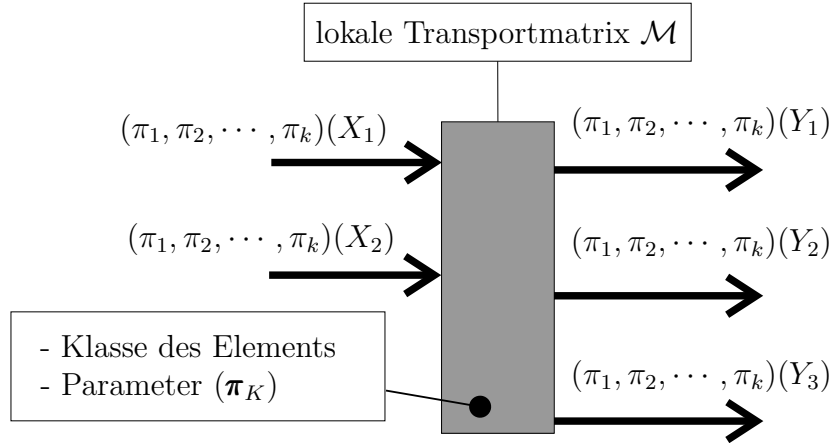
**Bemerkung:** Zwar könnten verschiedene Steuerregeln, wie das erwähnte Parkverhalten von Regalbediengeräten auch mit einem Parameter kodiert werden. Der Aufwand für die Integration solcher willkürlichen Werte in das numerische Modell ist jedoch unverhältnismäßig hoch. Zudem bleibt das Modell übersichtlicher, wenn für jede Steuerstrategie jeweils eine Klasse zur Verfügung steht.

Seien nun die Ein- und Ausgänge des Transportsystem-Elements betrachtet. Da die Verteilungsfunktionen „als solche“ nicht numerisch handhabbar (und auch selten gegeben) sind, müssen Parameter gesucht werden, die diese Funktionen möglichst gut repräsentieren.

Sei  $Z \in \{X_1, X_2, \dots, X_{n_E}, Y_1, Y_2, \dots, Y_{n_A}\}$  und  $\pi(Z) = (\pi_1, \pi_2, \dots, \pi_k)(Z)$  das k-Tupel der Parameter, die die Verteilungsfunktion  $F_Z(t)$  ( $t > 0 \in \mathbb{R}$ ) der Zeitabstände an einer Stelle  $Z$  beschreibt. Die Wahl der Parameter soll gleichermaßen für die Eingänge ( $X_i$ ) und die Ausgänge ( $Y_j$ ) gelten (Abbildung 4.3).

---

<sup>3</sup>Als Parken wird die Zeit bezeichnet, in der kein Auftrag für das Gerät vorliegt. Eine Möglichkeit ist, dass das Bediengerät dort wartet, wohin es sich für den jüngste Auftrag bewegt hat. Eine andere Variante besteht darin, dass es zum Einlagerungspunkt fährt und dort auf neue Aufträge wartet. Offensichtlich bevorzugt diese Strategie Einlagerungsaufträge.



**Abbildung 4.3:** Parametrisches Modell eines Transportsystem-Elements

Im einfachsten Fall beschränkt man sich auf die ersten Momente oder, äquivalent dazu, die mittlere Transportrate an dieser Stelle. Diese Daten sind in der Regel für jeden Ort vorhanden oder ermittelbar. Für eine Zufallsgröße  $Z$  von Zwischenzeiten wird der Erwartungswert von  $Z$

$$\pi_1(Z) := EZ = \int_0^{\infty} t \cdot dF_Z(t) \quad (4.2)$$

benutzt.

Der minimale Abstand zwischen zwei Transporteinheiten ist eine wichtige Größe zur Charakterisierung des Gutstromes. Der zweite Parameter in diesem Modell sei deshalb die minimale Zwischenzeit an der Stelle  $Z$ , also

$$\pi_2(Z) := \min(Z). \quad (4.3)$$

Im Folgenden soll die etwas aussagekräftigere Schreibweise  $t_{\min}(Z) := \min(Z)$  benutzt werden. Für die Ausgänge  $Y_j$  kann dieser Parameter aus den technischen und steuerungstechnischen Gegebenheiten des betrachteten Elements abgeleitet werden. Für die Eingänge  $X_i$  ist er aus dem Umfeld des Transportsystem-Elements herzuleiten. Oft ist jeder Eingang gleichzeitig Ausgang eines vorgelagerten Elements. Die minimale Zwischenzeit dessen Abgangsstrom ist aber bekannt. Für Forderungen, die von außen in das System getragen werden, sind entsprechende Abschätzungen zu treffen.

**Bemerkung:** Für Materialfluss-Systeme ist, technisch bedingt, diese Zeit  $\pi_2(Z)$  immer größer als Null. Mit  $t_{\min} > 0$  ist der Gutstrom nicht mehr markovsch. Untersuchungen auf der Basis der Warteschlangentheorie wären mit diesen Modellannahmen kaum noch handhabbar.

Eine maximale Zwischenzeit als Parameter zu benutzen, erscheint hingegen nicht sinnvoll, da diese kaum einen Einfluss auf das Verhalten des Transportsystem-Elements hat. Es ist an dieser Stelle hilfreich, kurz einen Blick auf den bisherigen

Modellbildungsprozess zu werfen. Die jeweiligen Modellannahmen bzw. Modellierungsschritte sind hier kurz dargestellt:

1. Beginn der Modellierung mit einem ereignisdiskretem Modell für ein Transportsystem-Element,
2. Ausschluss globaler Steuerung,
3. Annahme der Blockierungsfreiheit,
4. Stationarität der Prozesse,
5. Übergang von Prozessen zu statistischen Verteilungsfunktionen der Zwischenzeiten,
6. Klassierung und Parametrierung der Transportsystem-Elemente und der Verteilungsfunktionen.

Ziel des letzten strukturellen Schrittes, der Parametrierung, ist es, ein numerisches Modell eines Transportsystem-Elements zu schaffen, das dann mit Hilfe von neuronalen Netzen approximiert wird. Dabei wird für jeden zu berechnenden Parameter ein separates Netz benutzt.

In einem kompositen Modell sollen später (Kapitel 5) die an dieser Stelle entwickelten Modelle miteinander verbunden werden. Die Ausgänge des einen Elements sind dann Eingänge eines anderen. Das bedeutet, dass jeder Parameter, der dem Modell an einem Eingang zur Verfügung gestellt wird, auch für jeden der Ausgänge berechnet werden muss. Dieses Modell ist offen für beliebige weitere Parameter; eine Diskussion der gegebenenfalls sinnvollen Kenngrößen findet sich im Abschnitt 6.2.2.

Im Rahmen dieser Arbeit sollen nur die Variationskoeffizienten  $c_Z$  von  $Z$  hinzugenommen werden:

$$\pi_3(Z) := c_Z = \frac{\sqrt{D^2 Z}}{EZ}, \quad (4.4)$$

wobei

$$D^2 Z = \int_0^\infty (t - EZ)^2 \cdot dF_Z(t) \quad (4.5)$$

die Streuung von  $Z$  ist.

Durch diesen Schritt der Parametrierung wird über alle statistischen Verteilungsfunktionen mit gleichen Erwartungswerten, Minima und Streuungen abstrahiert. Unabhängig vom konkreten Verlauf werden ab jetzt zwei Verteilungsfunktionen mit gleichem Erwartungswert, Minimum und Variationskoeffizienten miteinander identifiziert.

### 4.1.5 Numerische Approximation mit neuronalen Netzen

An dieser Stelle der Modellierung sollen neuronale Netze eingesetzt werden: Erst eine gute und handhabbare numerische Approximation rechtfertigt die vorangegangenen Schritte der Modellbildung.

Zunächst seien die „interessierenden statistischen Kenngrößen“ betrachtet: In der bisherigen Modelldiskussion wurden sie aus dem Grunde vernachlässigt, weil sie keinen Einfluss auf den Gültigkeitsbereich des Modells haben. Während die Parametertupel, die die Ausgänge charakterisieren, für die Einbettung eines einzelnen Transportsystem-Elements in einen Verbund gleichartiger Modelle wesentlich sind, interessieren den Anwender des Modells doch gänzlich andere Kenngrößen, wie etwa

- Auslastung  $\varrho$  des Transportsystem-Elements oder
- Quantile der Warteschlangenlängen für die einzelnen Eingänge.

Diese Arbeit untersucht die Machbarkeit und die Nützlichkeit dieses neuen Modellierungsansatzes. Deshalb beschränkt sie sich an dieser Stelle darauf, für alle betrachteten Modell-Klassen die Auslastung und das 90%-Quantil  $Q_{90}$  der Warteschlangen an den Eingängen abzubilden. Aus ersten Experimenten kann aber bereits jetzt konstatiert werden, dass sich die gleichen neuronalen Netze auch für Prognosen anderer statistischer Parameter trainieren lassen. (Eine Diskussion über weitere Parameter findet in Abschnitt 6.2.2 statt.) Die **Zielgrößen** der Approximation sind damit die folgenden Parameter:

- Für jeden Ausgang  $Y_j$  ( $j = 1, 2, \dots, n_A$ ) des Elements sind die Parameter  $\pi_3$ , also die Variationskoeffizienten der Abgangsverteilung (Gleichung 4.4), zu approximieren. Die anderen Werte des Parametertupels  $\pi_1(Y_j) = EY_j$  und  $\pi_2(Y_j) = t_{min}(Y_j)$  ergeben sich auf analytischem Weg.
- Die Auslastung  $\varrho$  des Elements wird numerisch approximiert, sofern sie nicht analytisch berechenbar ist.
- Für jeden Eingang  $X_i$  ( $i = 1, 2, \dots, n_E$ ) des Elements ist das 90%-Quantil der Warteschlange zu approximieren.

Mit Hilfe der **Eingangsdaten** (Argumente) wird ein konkretes Transportsystem-Element in seinem logistischen Kontext definiert. Sie haben die Form

$$\mathcal{P} = (\pi_K, \pi(X_1), \pi(X_2), \dots, \pi(X_{n_E}), \mathcal{M}).$$

Der zusammengesetzte Parametervektor  $\mathcal{P}$  werde als *Gesamtparameter* bezeichnet. Dabei sind die einzelnen Komponenten des Vektors:

- der klassenspezifische Parametervektor  $\pi_K$ , der das konkrete Element beschreibt,
- die Parametertupel  $\pi(X_i)$  für jeden Eingangstrom  $X_i$  ( $i = 1, \dots, n_E$ ) und
- die lokale Transportmatrix  $\mathcal{M}$ , die über die Quelle-Ziel-Beziehungen innerhalb des Transportsystem-Elements Auskunft gibt.

### Warum neuronale Netze?

Die numerische Approximation der Modellparameter ist, wie schon erwähnt, ein Schritt, der über die Sinnfälligkeit des gesamten Modells entscheidet. Gleichwohl ist es nicht Ziel der vorliegenden Arbeit, verschiedene numerische Ansätze auf ihre Verwendbarkeit für dieses Problem zu untersuchen.

Nichtsdestotrotz seien hier einige Gründe aufgeführt, die zur Wahl des Werkzeugs „neuronale Netze“ geführt haben:

- Die Dimension des gesamten Parameterraums ist beträchtlich: Für die relativ einfache Modellklasse „Kreuzung“ werden 8 Dimensionen (bereits nach einer Parameterreduktion) benötigt. Für kompliziertere Elemente sind weit mehr Parameter erforderlich.
- Die Suche nach geeigneten numerischen Ansatzfunktionen gestaltet sich bei Räumen hoher Dimension und nicht-separierbaren Problemen schwierig. Neuronale Netze können deshalb fast als „Numerik für Einfallslose“ betrachtet werden.
- Der zu untersuchende Raum der Modellparameter kann so gewählt werden, dass die zu approximierenden Zielfunktionen über diesem Gebiet beschränkt und stetig differenzierbar sind. Das ermöglicht den Einsatz glatter Aktivierungsfunktionen (etwa der logistischen Funktion) in den Neuronen und vereinfacht damit die Arbeit mit den Netzen.
- Ansätze, die über orthogonalen Gittern arbeiten, müssten mit wenigen Gitterpunkten pro Dimension auskommen. Bereits für eine Dimension von 8 sind für fünf Gitterpunkte pro Dimension  $5^8 = 390625$  Stützpunkte notwendig. Für 5 s pro Parametertupel ergibt sich eine Rechenzeit von über drei Wochen. Ob jedoch 3 oder 4 Gitterpunkte pro Parameter ausreichen, hängt sicher vom konkreten Ansatz ab und kann hier nicht beurteilt werden. Folgerung ist jedoch, dass die Stützstellen für die Approximation zufällig im Parameterraum gewählt werden.
- Voruntersuchungen bestätigten, dass neuronale Netze imstande sind, bei relativ geringem Aufwand numerische Approximationen mit geringem Fehler zu liefern.
- Für das Training der Netze steht mit dem Stuttgarter Simulator ein praktisches Werkzeug zur Verfügung (siehe Anhang A.2, S. 91).
- Mit zusätzlichen Programmen kann das fertige neuronale Netz in einen C-Quelltext übersetzt werden und ist damit universell und unabhängig von diesem Simulator einsetzbar.

## Erzeugung der Trainingsdaten

Die Bereitstellung von Daten an den Stützstellen geschieht in der folgenden Weise:

1. Für eine Klasse von Transportsystem-Elementen wird ein diskretes Simulationsmodell gebaut (siehe Anhang A.1).
2. Aus dem Raum sinnvoller Parameter für das Element, für die Eingangsverteilungen und die lokale Transportmatrix wird zufällig ein Gesamtparameter  $\mathcal{P}$  ausgewählt. Dieser repräsentiert jetzt ein konkretes Transportsystem-Element mit einer konkreten Belastungssituation.

Für die Eingangsströme werden verschobene Gammaverteilungen benutzt. Für eine zugehörige Dichtefunktion  $f_Z(t)$  gilt:

$$f_Z(t) = \begin{cases} \frac{(t-t_{\min}(Z))^{\alpha-1} e^{-(t-t_{\min}(Z))/\beta}}{\beta^\alpha \Gamma(\alpha)} & \text{falls } t > t_{\min}(Z), \\ 0 & \text{sonst} \end{cases} \quad (4.6)$$

mit dem Formparameter (*shape*)  $\alpha$  und dem Skalierungsfaktor (*scale*)  $\beta$ . Dabei gilt

$$\alpha = c_Z^{-2} \quad \text{und} \quad \beta = EZc_Z^2. \quad (4.7)$$

**Bemerkung:** Die Überlagerung vieler unabhängiger Ereignisse ist recht gut mit einer Exponentialverteilung beschreibbar. Die Hintereinanderausführung zufälliger Bedienprozesse nähert sich einer Normalverteilung. Beide Extremfälle können mit Hilfe einer Gammaverteilung erfasst werden.

3. Eine ereignisdiskrete Simulation läuft über mehrere Simulationstage, um die entsprechenden Ergebnisse (Variationskoeffizienten der Abgangsverteilungen, Warteschlangen-Quantile der Eingänge, Auslastung) zu berechnen. Diese werden abgespeichert.
4. Solange, bis eine genügend große Anzahl von Gesamtparametern berechnet wurden, wird bei Schritt 2 fortgesetzt. (Aussagen zum Umfang der Trainingsdaten finden sich auf S. 43.)

Die Einschränkung des Parameterraums auf einen sinnvollen Bereich wird im Rahmen dieser Arbeit nur exemplarisch vollzogen. Während es für die technischen Parameter Referenzen gab, sind Variationskoeffizienten von Eingangsströmen für die verschiedenen Transportsystem-Elemente nur schwer zugänglich. Man kann aber annehmen (siehe Kapitel 5), dass im Wesentlichen jeder Eingang eines Elements mit dem Ausgang des vorgelagerten Elements verbunden ist. In Voruntersuchungen wurden deshalb für die Eingänge der Elemente Gutströme mit Variationskoeffizienten zwischen 0.1 und 2 benutzt. Es stellte sich heraus, dass die Abgangsströme im Wesentlichen<sup>4</sup> nur noch Variationskoeffizienten zwischen 0.2 und 1.4 hatten. Aus diesem

<sup>4</sup>Insbesondere bei sehr geringer Auslastung des Materialfluss-Elements sind die Variationskoeffizienten der Ausgänge stärker von denen der Eingänge bestimmt.



Grunde scheint es plausibel, auch für sämtliche Eingangsströme  $X$  anzunehmen, dass für deren Variationskoeffizienten gilt:

$$0.2 \leq c_X \leq 1.5 \quad .$$

Bei einigen Modellklassen wird die Auslastung  $0 < \varrho < 1$  zur Festlegung des sinnvollen Bereiches eines Parameters (etwa der mittleren Bedienzeit einer Bedienstation) benutzt. An dieser Stelle wäre eine Einflussnahme auf die verstärkte Berücksichtigung hoher Auslastungen möglich. Aufgrund der geringen praktischen Erfahrungen mit der vorgestellten Modellierung ist es jedoch noch zu früh, hier spezielle Verteilungen der Auslastungen anzunehmen.

Wenn möglich, sollte die Anzahl der Parameter reduziert werden. In den meisten Fällen hat man die Gelegenheit, die Dimension um Eins zu verringern, indem alle zeitlichen Parameter auf den minimalen Abstand  $0 < t_{min}$  zweier Ereignisse an einem Eingang normiert werden. Die Zeit wird für das Element dann nicht mehr in Sekunden sondern beispielsweise als Vielfaches von  $t_{min}$  angegeben.

Sei  $Z$  eine Zufallsgröße. Für  $\hat{Z} = t_{min}Z$  mit  $0 < t_{min} \in \mathbb{R}$  gilt für die Streuung

$$D^2\hat{Z} = D^2(t_{min}Z) = t_{min}^2 D^2Z. \quad (4.8)$$

Für den Variationskoeffizienten  $c_{\hat{Z}}$  ergibt sich dann:

$$c_{\hat{Z}} = \frac{\sqrt{D^2\hat{Z}}}{E\hat{Z}} = \frac{t_{min}\sqrt{D^2Z}}{t_{min}EZ} = c_Z. \quad (4.9)$$

Damit sind die Variationskoeffizienten invariant gegenüber dieser Transformation der Zeit.

Die Festlegung der einzelnen Parameter wird jeweils in den Abschnitten „Modelldaten“ für die konkreten Modellklassen beschrieben.

Der Umfang der Daten hängt von verschiedenen Nebenbedingungen ab: Dimension des Problems, Größe des neuronalen Netzes, Aussehen der zu approximierenden Hyperfläche, verfügbare Rechenzeit. Er muss von Fall zu Fall aufs Neue entschieden werden. Bei den hier betrachteten Fällen war es nötig, je nach Dimension mit etwa 2000 bis 3000 Stützstellen zu arbeiten. Denn schon beim Training mit nur 1500 Datensätzen konnte für einige Modellklassen ein „Übertrainieren“ (*overfitting*) beobachtet werden. In diesem Fall wurde zwar der Trainingssatz gut approximiert, aber für andere Stützstellen versagte das Netz.

Für die betrachteten Modellklassen wurde die Dauer der Simulation dynamisch in Abhängigkeit von der Güte der empirischen statistischen Daten bestimmt: Wenn sich die empirischen 90%-Quantile der Warteschlangen und Variationskoeffizienten innerhalb eines Simulationstages nur noch um weniger als 0.01 änderten, wurde die Genauigkeit als ausreichend anerkannt und die Simulation beendet.

## Topologie der neuronalen Netze

Die Anzahl der Eingangsneuronen des Netzes ist durch die Dimension des Problems gegeben. Es hat sich gezeigt, dass es günstiger ist, für jeden Zielparameter (vgl. S. 40) ein eigenes Netz zu trainieren. Ein Grund dafür liegt in der unterschiedlichen Konvergenzgeschwindigkeit und unterschiedlichen Güte der Approximationen für die einzelnen Parameter.

Es gibt zwei Wege, die Topologie eines neuronalen Netzes zu verbessern. Der eine besteht in der Vergrößerung, der andere in der Verkleinerung einer bestehenden Struktur. Falls der Fehler des neuronalen Netzes nach dem Trainingsprozess noch zu groß ist, kann es daran liegen, dass das Netz nicht in der Lage ist, die interessierende Funktion gut zu approximieren. Die Hinzunahme weiterer Neuronen erhöht den Freiheitsgrad des Netzes und damit die Möglichkeit, bessere Ergebnisse zu liefern. Ein anderer Weg besteht darin, ein zu großes Netz zu reduzieren, bis eine gute Balance zwischen Trainingszeit und Genauigkeit erreicht wird.

Es kann keine allgemeine Vorschrift zum Design der Topologie eines neuronalen Netzes angegeben werden [Moo94]. Der Entwurf eines Netzes ist ein empirischer Weg, bei dem aber mit dem Simulator SNNS (Anhang A.2) ein gutes Hilfsmittel zur Verfügung steht.

Wenn sich eine Topologie für ein Problem bewährt hat, wird sie nach Möglichkeit wiederverwendet. Für die Prognosen der Variationskoeffizienten und der Auslastung eines Transportsystem-Elements war das immer möglich. Die Warteschlangenquantile sind schwerer zu approximieren, das liegt zum einen daran, dass sie für höhere Auslastung des Elements eine starke Steigung erfahren, zum anderen daran, dass gerade für diese Bereiche weniger Trainingsdaten zur Verfügung stehen. Spätere Untersuchungen können gegebenenfalls von der Gleichverteilung der Parameter-Stützstellen im Parameterraum abgehen, um gezielt sehr hohe Warteraumbelegungen zu betrachten. Da bislang keine Daten über Häufigkeiten einzelner Parameter vorliegen, werden sie im Rahmen dieser Arbeit für die nachfolgenden Modellierungen als gleichverteilt vorausgesetzt.

Für jede berücksichtigte Modellklasse werden im entsprechenden Abschnitt die benutzten Netze mit ihren Eingangs- und Abgangsgrößen abgebildet. Mit Sicherheit können die neuronalen Netze noch optimiert werden, dafür gab es jedoch keinen akuten Handlungsbedarf. Da die Netze relativ klein sind, ergeben sich geringe Trainings- und Rechenzeiten.

## Transformation der Trainingsdaten

Vor dem Training der Netze werden Parametertransformationen vorgenommen. Die Zielwerte müssen so normiert werden, dass sie zwischen 0 und 1 liegen. In allen Modellen werden deshalb zum Training der Netze als Zielgrößen  $c_Z/2$  bzw.  $Q_{90}(Z)/10$  benutzt. Variationskoeffizienten größer als zwei wurden nicht beobachtet, Warteschlangenquantile größer als 10 erscheinen technisch nicht sinnvoll.

Mit der Einbettung in entsprechende Software, die die Parametertransformationen (und Rücktransformationen) übernimmt, ist der Modellbildungs-

prozess abgeschlossen. Ergebnis an dieser Stelle ist ein numerisches Modell eines Transportsystem-Elements, das zu einer gegebenen Konfiguration von fördertechnischen und logistischen Parametern vorhersagt, welche 90%-Quantile die Warteschlangen an den Eingängen, welche Variationskoeffizienten an den Ausgängen des Elementes zu finden sind und wie groß dessen Auslastung ist.

Ein Modell besitzt jedoch erst eine Existenzberechtigung, wenn es „taugt“. Nur im Ausnahmefall werden in der Praxis die an einem Element eintreffenden Transporteinheiten gammaverteilte Eingangszeiten haben. Die Eingangsverteilungen bleiben normalerweise auch verborgen, so dass sich die Frage stellt, ob man dieses Modell trotzdem nutzen kann, und mit welchen Fehlern man in diesem Fall zu rechnen hat.

### 4.1.6 Gültigkeit und Korrektheit

Robinson [Rob97] betont, dass das Ziel von Validierung und Verifikation eines Modells darin besteht, das Vertrauen zu stärken. Dies kann geschehen, indem nachgewiesen wird, dass das Modell in sich stimmig und fehlerfrei ist (Verifikation), und dass es entsprechend den Ansprüchen seiner Nutzer eingesetzt werden kann (Validierung) [Car86].

Bereits bei der Beschreibung der Modellbildung wurde Wert darauf gelegt, zu betonen, welche Eigenschaften auf der jeweiligen Abstraktionsstufe nicht mehr berücksichtigt werden können. Die Korrektheit des Modells ließe sich beweisen, indem man nachwiese, dass für alle Parameterkonfigurationen das neuronale Netz dieselben Zielgrößen berechnet, wie sie durch eine diskrete Simulation ermittelt werden. Das ist natürlich nicht möglich. Statt dessen wird hier auf eine statistische Argumentation zurückgegriffen.

### Allgemeine Konfidenzabschätzungen

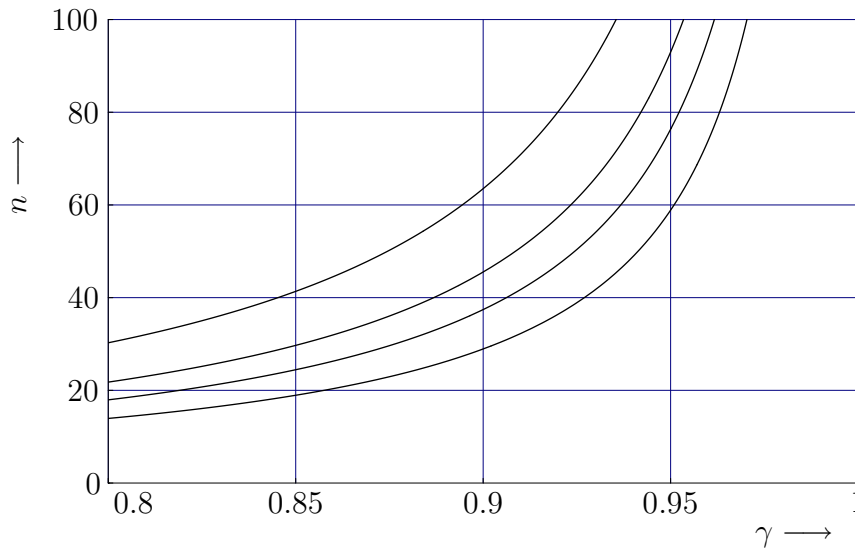
Einen allgemeinen Zusammenhang zwischen dem Umfang  $n$  einer Stichprobe und dem Konfidenzintervall einer dabei ermittelten Statistik einer Zufallsgröße  $X$  findet man zum Beispiel bei [Ras76] oder [Sto01]: Für den minimalen Umfang  $n_0$  einer Stichprobe, bei dem mit einer Wahrscheinlichkeit  $p$  ein Anteil  $\gamma$  ( $= \gamma \cdot 100\%$ ) der Grundgesamtheit von  $X$  zwischen dem Minimum  $x_{min}$  und dem Maximum  $x_{max}$  der Stichprobe liegt, gilt:

$$n \cdot \gamma^{n-1} - (n-1)\gamma^n \leq 1 - p. \quad (4.10)$$

Lösungen dieser Gleichung finden sich in Tafeln der Statistik. Abbildung 4.4 skizziert die Abschätzung des minimalen Stichprobenumfanges für ausgewählte Werte der Wahrscheinlichkeiten  $p$ .

Für einen Stichprobenumfang von 25 kann man also sagen, dass für einen Anteil von  $\gamma = 85\%$  der Grundgesamtheit mit einer Wahrscheinlichkeit von  $p = 90\%$  die Werte zwischen dem Minimum ( $x_{min}$ ) und dem Maximum ( $x_{max}$ ) der Stichprobe liegen.

Will man diese Konfidenzabschätzung auf die Verteilung des Fehlers der in diesem Kapitel beschriebenen Modelle anwenden, geht man folgendermaßen vor: Man



**Abbildung 4.4:** Stichprobenumfänge  $n$  für  $p=0.99, 0.95, 0.90, 0.80$  (von oben nach unten) nach Gleichung 4.10

erzeugt  $n$  Fehler-Stichproben vom Umfang  $n_s$ . Dazu müssen  $n \cdot n_s$  Simulationen durchgeführt werden, um den Betrag des Fehlers der Prognosen zu ermitteln. Für jede der Stichproben  $i = 1, 2, \dots, n$  berechnet man ihr Quantil<sup>5</sup>  $Q_q^i$  der Ordnung  $q$ . Betrachtet man jetzt diese Quantile als Realisierungen einer Zufallsgröße, die mit einem Umfang von  $n$  ermittelt wurden, so können für  $0 < q \leq 1$  nach der Abschätzung (4.10) Konfidenzgrenzen dafür angegeben werden, dass für den Großteil der Grundgesamtheit das Quantil  $q$ -ter Ordnung zwischen dem kleinsten und dem größten beobachteten Quantil der Ordnung  $q$  liegt.

Als Beispiel wird die Fehlerverteilung für das 90%-Quantil der Warteschlange vor einem Regalbediengerät betrachtet. (Eine ausführliche Beschreibung dieser Modellklasse findet man auf S. 64 ff.) Abbildung 4.5 zeigt für 1000 Experimente die kumulative relative Häufigkeit<sup>6</sup> der Absolutbeträge der Fehler  $\epsilon := |Q_N - Q_S|$ . Dabei sind  $Q_S$  und  $Q_N$  zueinander gehörige Quantilwerte aus der Simulation bzw. aus der Modellierung mit einem neuronalen Netz. Man erkennt, dass das Modell mit ausreichender Sicherheit fehlerfrei ist, d.h.  $\epsilon \ll 0.5$ .

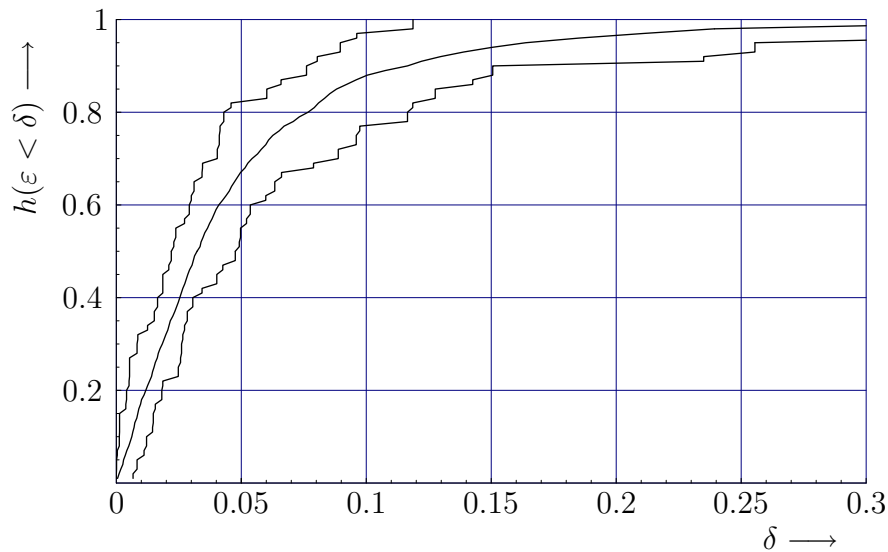
Es seien  $n = 37$  und  $n_s = 40$ . Das zugehörige Konfidenzintervall liefert mit  $n \cdot n_s = 37 \cdot 40 = 1480$  Simulationsexperimenten nach Gleichung (4.10) bzw. Abbildung 4.4 eine 90%-ige Sicherheit ( $p$ ) für 90% aller Werte ( $\gamma$ ).

Sowohl das Konfidenzniveau als auch die Größe des Konfidenzintervalls sind mit einem solch kleinen Stichprobenumfang nicht geeignet, die Aussage der zufälligen Verteilung des Fehlers aufzuwerten.

Ein weiterer Grund lässt von der Angabe von Konfidenzgrenzen absehen: Die Simulationen können nur für vorher definierte Eingangsverteilungen durchgeführt

<sup>5</sup>Die hier erzeugten Stichproben sind zu klein, um unbedenklich die verstetigten Quantile von S. 15 zu benutzen. Deshalb wird in Abbildung 4.5 die diskrete Form – vgl. linke Seite von Gleichung (2.1) – benutzt.

<sup>6</sup>Da es sich hierbei um die Auswertung eines Zufallsexperiments handelt, kann in der Abbildung nicht  $P(\epsilon < \delta)$  geschrieben werden, sondern man benutzt die relative Häufigkeit  $h(\epsilon < \delta)$ .



**Abbildung 4.5:** Verteilung und Konfidenzintervall des Fehlerbetrages  $\varepsilon$  der Approximation des 90%-Quantils der Warteschlange vor einem Regalbediengerät

werden. Die Vielfalt der in der Praxis auftretenden Fälle kann nicht berücksichtigt werden, so dass hier mit der Angabe von Konfidenzintervallen ein Vertrauen in die statistischen Daten suggeriert würde, das in dieser Form nicht zu rechtfertigen ist.

### Verifikation der Modelle

Der Fehler in einer zufälligen hinreichend großen Stichprobe von Parametertupeln ist genauso verteilt wie über dem gesamten Parameterbereich, so dass man bei Stichproben mit 500 - 1500 Parametertupeln eine recht gute Näherung der Fehlerverteilung für den gesamten Raum erhält. Es werden für Verifikation und Validierung deshalb die zufälligen Verteilungen der Fehler als Näherungen der tatsächlichen Fehlerverteilungen für die entsprechenden Eingangsverteilungen benutzt (siehe Anhang B, S. 96 ff.).

Werden für das Training von neuronalen Netzen nicht ausreichend Daten bereitgestellt, kann es geschehen, dass die Netze zwar auf den Stützstellen gute Ergebnisse liefern, aber für andere Eingabewerte versagen (*overfitting*) [Sch97]. Aus diesem Grund gehören Training und Test bei neuronalen Netzen zusammen. Denn wegen der hohen Dimension der Approximationsaufgabe ist etwa eine graphische Überprüfung der Richtigkeit nicht möglich.

Mit einer Menge von bislang nicht genutzten Daten aus den gleichen Simulationen wird deshalb jeweils gesichert, dass die neuronalen Netze nicht nur für ihre Trainingsdaten gute Ergebnisse liefern. In den entsprechenden Abschnitten und im Anhang B finden sich konkrete Aussagen für die betrachteten Modellklassen.

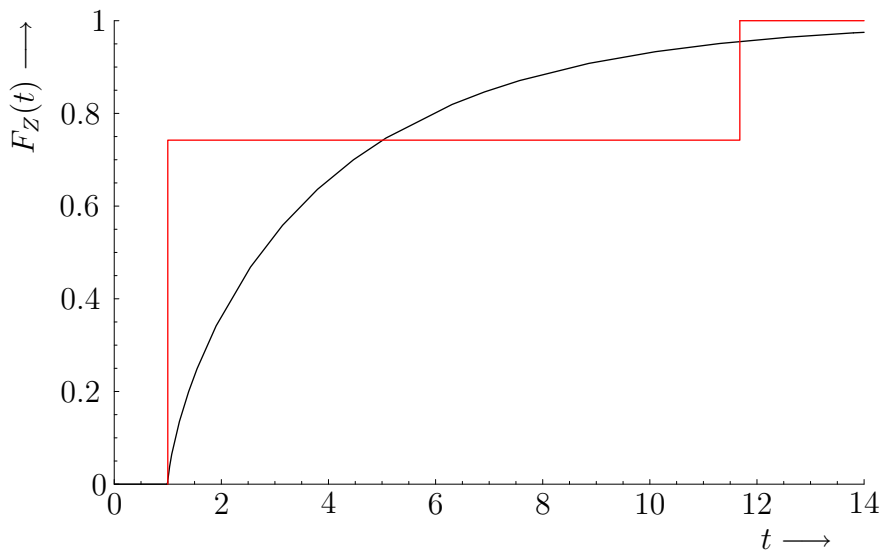
Aus der Sichtweise der neuronalen Netze ist das eine Validierung, also ein „Vertrauensbeweis“ dafür, dass sie die Größen im gesamten Parameterraum nähern können. Aus der Sicht eines Klassenmodells ist dieser Schritt eine Verifikation, denn er soll sichern, dass das Modell richtig *rechnet*.

Den einzelnen Ergebnissen vorausgreifend kann für alle Modellklassen festgestellt werden, dass die Variationskoeffizienten  $c_Y$  der Abgangsverteilungen sehr gut approximiert werden können. Das ist eine wesentliche Voraussetzung für die Verkettung solcher Modelle. Das Gleiche gilt auch für die numerischen Berechnungen der Auslastungen  $\rho$  der Transportsystem-Elemente.

Die Quantile der Warteschlangenlängen sind jedoch, selbst bei langer Laufzeit, mehr von den konkreten Simulationsläufen abhängig und streuen daher stärker als die anderen statistischen Größen.

## Validierung

Ziel der Validierung ist es, erfolglos zu zeigen, dass das Modell nicht stimmt [Rob97]. Nur dann, wenn man die Grenzen des Modells absteckt, kann ein Vertrauen in die Zuverlässigkeit in seinem Kernbereich geweckt werden. Ein erster Validierungsschritt für das Modell ist es somit, für die Eingangsverteilungen des Elements eine andere Verteilung als die Gammaverteilung zu benutzen.<sup>7</sup> Dafür muss eine möglichst einfache statistische Verteilung angegeben werden, die sich „möglichst stark“ von der Gammaverteilung unterscheidet<sup>8</sup>. Die Wahl fiel auf die Zweipunktverteilung. Abbildung 4.6 zeigt zur Veranschaulichung eine verschobene Gamma- und eine Zweipunkt-Verteilung mit gleichen Variationskoeffizienten ( $c_Z = 1.2$ ), gleichem Erwartungswert ( $EZ = 4$ ) und dem minimalen Abstand  $t_{\min}(Z) = q$ .



**Abbildung 4.6:** Verschobene Gammaverteilung (fett) und Zweipunkt-Verteilung mit gleichen Parametern ( $EZ = 4, c_Z = 1.2, t_{\min}(Z) = 1$ )

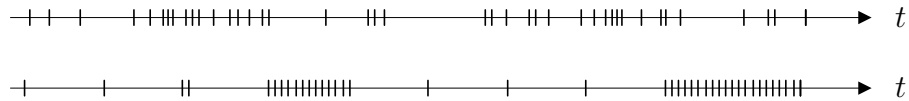
Für jede Modellklasse werden deshalb zur Validierung Simulationsläufe mit zweipunktverteilten Eingangsprozessen, durchgeführt. Die Unterschiede zwischen An-

<sup>7</sup>Weitere Schritte zur Validierung bestehen im Testen des Modells. Insbesondere die Kombination der einzelnen Teilmodelle zu einem zusammengesetzten Modell (Kapitel 5) bietet dafür sehr gute Möglichkeiten.

<sup>8</sup>Die Wahl einer Verteilungsfunktion ist, wie der gesamte Validierungsprozess, stark subjektiv. Sie soll deshalb hier nicht gerechtfertigt, sondern plausibel gemacht werden.

kunftsprozessen mit Gamma- und Zweipunktverteilung verdeutlicht Abbildung 4.7. Dort werden Ausschnitte mit Ankunftsprozessen dieser beiden Verteilungen skizziert.

Die Zweipunktverteilung als Referenzverteilung zu benutzen, ist offensichtlich eine Herausforderung für dieses Modell. Denn ihr Ankunftsverhalten ist eher einem *Batch*-Betrieb zuzuordnen, der bei der Modellbildung recht frühzeitig ausgeschlossen wurde, da dabei die Belastung sehr ungleichmäßig verteilt ist.



**Abbildung 4.7:** Veranschaulichung von Ankunftsprozessen mit gleichen Parametern: Gammaverteilung (oben) und Zweipunkt-Verteilung (unten)

Im Folgenden werden die einzelnen hier untersuchten Modellklassen betrachtet. Für die erste Modellklasse werden Verifikation und Validierung ausführlich diskutiert. Da sich die Fehler bei allen betrachteten Elementen in ähnlicher Weise verhalten, wird jedoch bei den anderen Modellklassen auf diese detaillierte Behandlung verzichtet. Die Fehlerdiagramme sind im Anhang B angefügt.

## 4.2 Modellklassen

Eine Systematik von Transportsystem-Elementen wurde auf S. 16 vorgestellt. Die Klassierung von Transportsystem-Element nach der Anzahl ihrer Ein- und Ausgänge eignet sich jedoch für die hier dargestellte Modellierung nicht oder nur bedingt:

- Gudehus benutzt seine Klassierung [Gud99], um mittlere Streckenbelastungen zu modellieren. So wird ein Lager dort gleichzeitig als Quelle (Auslagerungen) und als Senke (Einlagerungen) betrachtet.  
Geht es jedoch um die Abbildung von Warteschlangen vor dem Lager kann dieses Modell nicht benutzt werden. Grund dafür ist, dass die Länge einer Warteschlange vor dem Lager auch von den Auslagerungsprozessen abhängt.
- Ein weiteres Beispiel, bei dem die Modellierung von Gudehus nicht angebracht ist eine Kreuzung mit zwei Eingängen. Während dort zwischen Elementen der Typen (2,1) und (2,2) unterschieden wird, ist es für die hier entwickelten Modelle unerheblich, ob dieses Element tatsächlich einen oder zwei Ausgänge hat. Entscheidend ist hier die lokale Transportmatrix des Elements. Darüber hinaus wird in jener Klassierung nicht die lokale Steuerung des Elements berücksichtigt.

Die Anzahl der Ein- und Ausgänge ist nur *ein* Merkmal eines Transportsystem-Elements. Im Rahmen dieser Arbeit werden zur Klassierung der Elemente zusätzlich die technischen Parameter und die Steuerungsstrategie des Elements benutzt. Es werden die folgenden Klassen modelliert:

**Einschleusung:** Element vom Typ (2,1) mit stetiger Abfertigung,

**Ausschleusung:** Element vom Typ (1,2) mit stetiger Abfertigung,

**Kreuzung:** Element vom Typ (2,2) mit teilstetiger Abfertigung,

**Regalbediengerät:** bedingt als Element vom Typ (2,2) anzusehen, wobei ein Ein- und ein Ausgang Daten-Schnittstellen sind (Auslösen einer Auslagerung bzw. Bestätigung der Einlagerung),

**Bedienstation:** Element vom Typ (1,1).

Im Rahmen weiterer Arbeiten können auf dem hier dargestellten Weg noch weitere Transportsystem-Elemente mit Hilfe von neuronalen Netzen modelliert werden. Der Typ der Elemente kann dann als *ein* Merkmal zur Systematisierung genutzt werden.

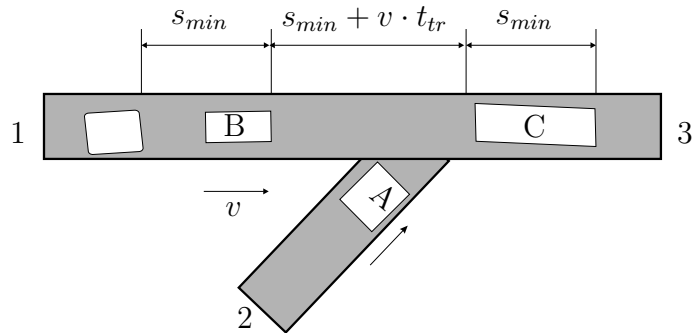


### 4.2.1 Einschleusung

#### Beschreibung des Transportsystem-Elements

Als einfachstes Beispiel für eine Zusammenführung wird im Rahmen dieser Arbeit die *Einschleusung* mit absoluter Vorfahrt [Gro84] der folgenden Art betrachtet:

Eine Nebenstrecke (Eingang 2) trifft auf die Hauptstrecke, die von Eingang 1 zum Ausgang 3 führt. Die Hauptstrecke habe die Geschwindigkeit  $v$ . Die Zeit für den Transfer einer Transporteinheit sei  $t_{tr}$  (siehe Abbildung 4.8).

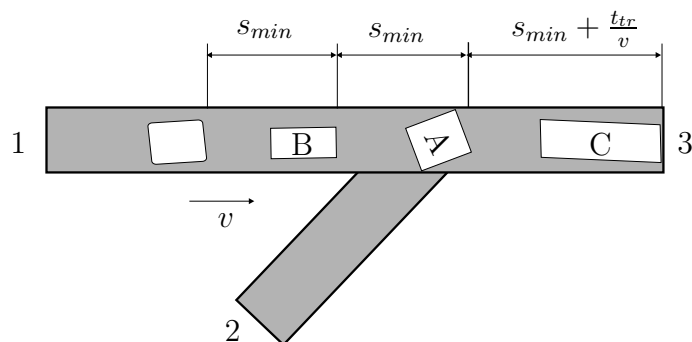


**Abbildung 4.8:** Einschleusung. Beginn des Transfers der Transporteinheit A

Aufgabe der Steuerung ist es, die in der Nebenstrecke ankommenden Transporteinheiten in die Lücken der Hauptstrecke zu platzieren. Dabei gilt:

- Die Hauptstrecke darf nicht angehalten werden.
- Der Mindestabstand zwischen zwei Transporteinheiten (Vorderkante zu Vorderkante) darf eine gewisse Länge  $s_{min}$  nicht unterschreiten.

Abbildung 4.9 zeigt die Situation nach erfolgreicher Einschleusung. Nur dann, wenn der Abstand zwischen zwei aufeinanderfolgenden Objekten der Hauptstrecke (im Beispiel B und C) größer als  $2s_{min} + v \cdot t_{tr}$  ist, kann ein Objekt der Nebenstrecke (A) eingeschleust werden.



**Abbildung 4.9:** Ende des Transfers

Unerheblich für die Modellierung ist die konkrete technische Ausführung (z.B. Transfer auf Gurtband, Rollenförderer oder in Kippschalen [Jün00]).

## Modelldaten

Wie bei der Modellbildung betont wurde (siehe S. 32), interessieren hier nur die Zeiten von Ereignissen. Problemlos lassen sich die Parameter dieses Transportsystem-Elements in zeitliche Kenngrößen transformieren. Nach dem Übergang

$$s_{min} \rightarrow t_{min} = s_{min}/v$$

sind die folgenden Daten zur Beschreibung der Einschleusung geeignet:

- Mindestzeitabstand (Vorderkante zu Vorderkante)  $t_{min}$  zwischen aufeinander folgenden Transporteinheiten (Hauptstrecke),
- Zeit  $t_{tr}$  für den Transfer einer Transporteinheit von der Nebenstrecke auf die Hauptstrecke,
- mittlerer Abstand  $EX_1$  zwischen den Ankunftszeiten aufeinander folgender Transporteinheiten auf der Hauptstrecke,
- mittlerer Abstand  $EX_2$  zwischen den Ankunftszeiten aufeinander folgender Transporteinheiten auf der Nebenstrecke,
- Variationskoeffizient  $c_{X_1}$  der Ankunftszeiten der Hauptstrecke,
- Variationskoeffizient  $c_{X_2}$  der Ankunftszeiten der Nebenstrecke.

Die Mindestabstände auf der Haupt- und Nebenstrecke seien gleich. Das erlaubt es, weitere Parameter zu reduzieren. Wegen Gleichung (4.9, S.43) können die auftretenden Zeiten auf  $t_{min}$  normiert werden. Alle Zeiten werden deshalb ab hier mit der Einheit  $t_{min}$  angegeben.

### Parameterraum für die Simulationen

Die sinnvollen Wertebereiche der Parameter werden mit Hilfe der Auslastung  $\varrho$  des Elements bestimmt:

Sei  $n = n_H + n_N$  die Summe der im Mittel in einer Zeit  $T$  abgefertigten Transporteinheiten auf der Hauptstrecke  $n_H = T/EX_1$  und der Nebenstrecke  $n_N = T/EX_2$ .  $T_G$  sei die Zeit, die insgesamt für deren Transport benötigt wird.

$$T_G = n_H t_{min} + n_N (t_{min} + t_{tr}) \quad (4.11)$$

Es seien

$$a = \frac{n_H}{n} \quad \text{bzw.} \quad 1 - a = \frac{n_N}{n}. \quad (4.12)$$

Dann ist offenbar

$$a EX_1 = (1 - a) EX_2. \quad (4.13)$$

Aus  $\varrho = \frac{T_G}{T}$  folgt mit (4.11)

$$\varrho = \frac{n_H t_{min} + n_N (t_{min} + t_{tr})}{T} \quad (4.14)$$

$$= \frac{t_{min}}{EX_1} + \frac{t_{min} + t_{tr}}{EX_2}. \quad (4.15)$$

Und wegen (4.12) erhält man

$$EX_1 = \frac{a \cdot t_{min} + (1 - a)(t_{min} + t_{tr})}{a \cdot \varrho} \quad \text{bzw.} \quad (4.16)$$

$$EX_2 = \frac{a \cdot t_{min} + (1 - a)(t_{min} + t_{tr})}{(1 - a) \cdot \varrho}. \quad (4.17)$$

Als sinnvolle Parameter werden  $0 < a < 1$  und  $0 < \varrho < 1$  benutzt.  $EX_1$  und  $EX_2$  ergeben sich entsprechend. Die Variationskoeffizienten in allen Modellen liegen zwischen 0.2 und 1.5 (siehe S. 43).

Für die Einschleuszeit wird  $0 \leq t_{tr} \leq 2$  angenommen. Eine Einschleuszeit von 0 bedeutet nicht, dass für den Transfer keine Zeit benötigt wird, sondern dass es für das Einschleusen ausreicht, auf der Hauptstrecke eine Lücke von  $2 \cdot t_{min}$  bzw.  $2 \cdot s_{min}$  zu finden. Das kann insbesondere bei Förderern mit festen Abständen oder bei spitzwinkligen Transfers der Fall sein.

Nachdem der Parameterraum festgelegt ist, können die Simulationsläufe beginnen. Für das Training der Netze erwiesen sich etwa 3000 Parametertupel<sup>9</sup> zuzüglich 1000 Tupel zur Verifikation als ausreichend.

#### *Parameter für das Netztraining*

Als transformierte Parameter für das Netztraining werden dann  $\frac{1}{EX_1} + \frac{1}{EX_2}$  (als Parameter für die mittlere Last)<sup>10</sup>,  $a$  (Anteil der Hauptstrecke an der Gesamtlast),  $t_{tr}$ ,  $c_{X_1}$  und  $c_{X_2}$  benutzt. Für jeden Parameter konnte ein neuronales Netz der folgenden Topologie genutzt werden. Die Eingabeschicht besteht aus fünf Neuronen, die Ausgabeschicht aus einem Neuron. Jede der fünf verdeckten Schichten ist komplett mit der vorhergehenden und nachfolgenden Schicht vernetzt. Zusätzlich existieren einige Verbindungen über mehrere Schichten hinweg (siehe Abbildung 4.10).

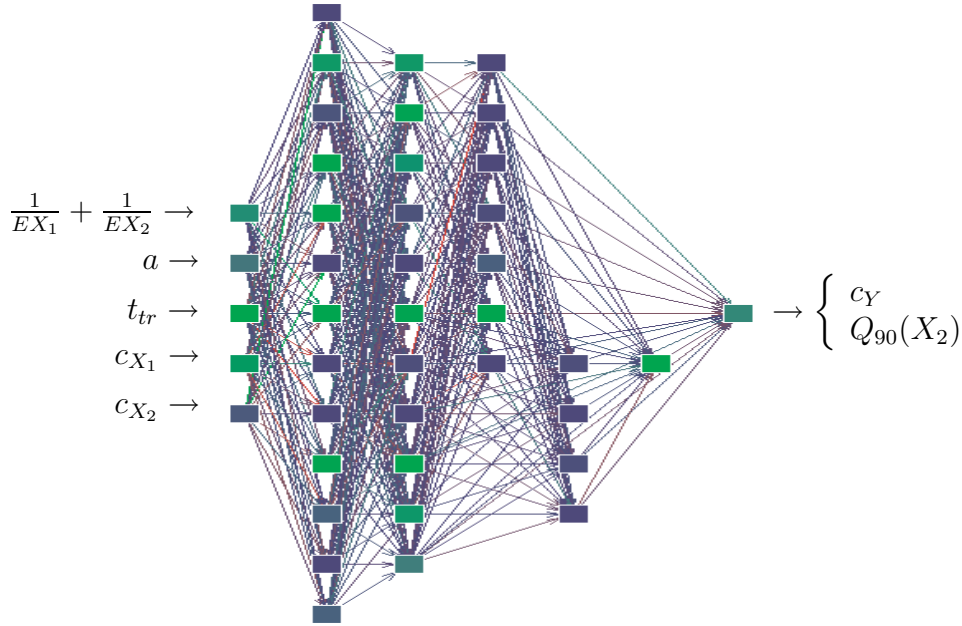
Bei diesem, wie auch bei den neuronalen Netzen der folgenden Abschnitte lesen sich die Abbildungen der jeweiligen Netzstruktur wie folgt: Links des Netzes findet man die Eingangsparameter. Am Ausgang des Netzes, hinter einer geschweiften Klammer, stehen die einzelnen Ausgangswerte des Netzes. In diesem Fall der Einschleusung werden unabhängig voneinander zwei neuronale Netze mit jeweils 5 Eingängen und einem Ausgang trainiert. Ein Netz berechnet den Variationskoeffizienten  $c_Y$ , das andere berechnet das Quantil  $Q_{90}(X_2)$ . Beide Netze haben dieselbe Topologie, deswegen werden sie in *einer* Abbildung dargestellt.

Die hier angegebene Topologie (Abbildung 4.10) ist nur *eine* mögliche, sie wurde auf heuristischem Wege (siehe S. 44) erzeugt. Die Abbildung soll nicht dazu anregen, eine identische Struktur nachzubauen, es soll lediglich die Komplexität des Netzes veranschaulicht werden.

**Bemerkung:** Die Anzahl der Neuronen in der Ein- und Ausgabeschicht wird durch die Anzahlen der Eingangs- bzw. Ausgangswerte bestimmt. Zu Beginn der Experimente wurde eine erste versteckte Schicht mit 13 Neuronen vollständig mit den Eingängen und dem Ausgang vernetzt, doch das Ergebnis der Approximation war nicht zufriedenstellend.

<sup>9</sup>Die Rechenzeit für 10 Tage Simulation beträgt etwa 5 s (s.a. Bemerkung auf S. 36).

<sup>10</sup>Diese Normierungen haben sich bewährt und spiegeln die Struktur des Problems besser wieder.



**Abbildung 4.10:** Einschleusung. Topologie der benutzten neuronalen Netze

Daraufhin wurden die beiden nächsten Schichten hinzugefügt. Doch auch das lieferte noch nicht befriedigende Ergebnisse, so dass die beiden letzten versteckten Schichten *zusätzlich* zur bestehenden Struktur eingefügt. Es wurden hier auch Kurzschlüsse, also Verbindungen über mehrere Schichten hinweg, zugelassen. Damit schließlich konnten gute Trainingsergebnisse erzielt werden.

Auf eine schrittweise Reduzierung der Neuronen und Verbindungen wurde verzichtet, da es im Rahmen dieser Arbeit nicht um die Ermittlung eines *minimalen* Netzes geht, sondern um prinzipielle Machbarkeit der vorgestellten Modellellierung.

Die Arbeit mit neuronalen Netzen wird durch den Stuttgarter Simulator für neuronale Netze (SNNS) erheblich erleichtert. Im Anhang A.2 findet man eine kurze Vorstellung des Programmpakets.

## Verifikation

Unterschiedliche Einsatzfälle bringen auch verschiedene Anforderungen an die Zuverlässigkeit der Modelle mit sich. Aus diesem Grund werden hier nicht nur prägnante Kennzahlen angegeben, sondern alle Ergebnisse im Anhang angefügt.

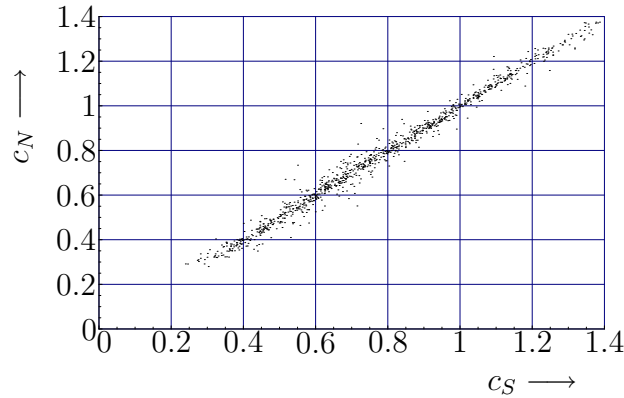
Zur Verifikation werden für 1000 Parameterkonfigurationen die Berechnungen des neuronalen Netzes mit den entsprechenden Simulationsdaten (gammaverteilte Eingänge) verglichen.

### Variationskoeffizient der Abgangsverteilung

Abbildung 4.11 zeigt die Beziehung zwischen den „realen“ Werten aus der Simulation ( $c_S$ ) und den Ergebnissen des neuronalen Netzes ( $c_N$ ) für die Variationskoeffizienten des Abgangsstromes.<sup>11</sup> Die Simulation wurde als ausreichend angesehen, wenn

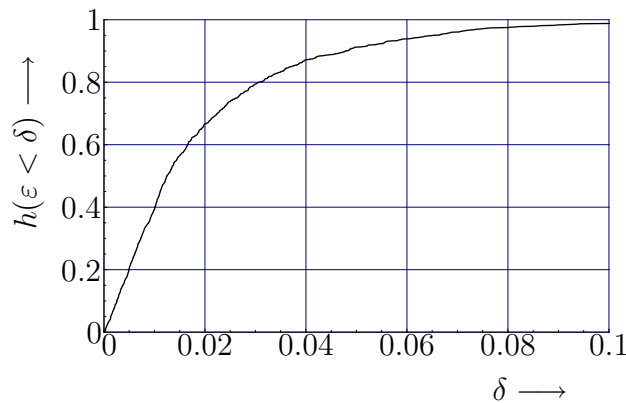
<sup>11</sup>Korrekterweise müssten in diesen Diagrammen  $c_Y^S$  bzw.  $c_Y^N$  benutzt werden. Da aber jeweils nur

sich nach einem Simulationstag die Warteschlangenquantile nur noch um höchstens 0.01 verändert haben. Die empirisch ermittelten Variationskoeffizienten veränderten sich dabei aber nur noch um etwa 0.001. Deren Genauigkeit ist also deutlich höher zu bewerten.



**Abbildung 4.11:** Einschleusung, Variationskoeffizient des Ausgangs.  
Vergleich der Netzdaten  $c_N$  mit den Simulationsdaten  $c_S$

Als Fehler wird für Verifikation und Validierung der Absolutbetrag der Differenz zwischen dem Simulationsergebnis und der zugehörigen Prognose  $\varepsilon = |c_N - c_S|$  benutzt. Aus der Verteilung<sup>12</sup> der Fehler (Abbildung 4.12) ist ersichtlich, dass die Abweichungen der vom neuronalen Netz prognostizierten Variationskoeffizienten von den simulativ ermittelten Werten marginal sind.



**Abbildung 4.12:** Einschleusung, Variationskoeffizient des Ausgangs.  
Verteilung der Netzfehler  $\varepsilon$

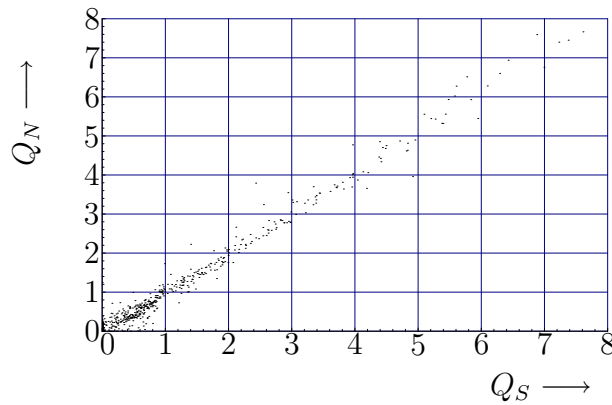
---

ein Variationskoeffizient dargestellt wird, soll der zweite Index aus Gründen der Übersichtlichkeit eingespart werden.

<sup>12</sup>Da es sich hierbei um die Auswertung eines Zufallsexperiments handelt, kann nicht  $P(\varepsilon < \delta)$  geschrieben werden, sondern es wird die relative Häufigkeit  $h(\varepsilon < \delta)$  benutzt. Während die Fehler der Approximationen von Variationskoeffizienten in der Größenordnung von 0.05 liegt, bewegen sie sich bei Warteschlangenquantilen um 0.5. Wesentlich genauere Approximation ist für beide Fälle nicht erforderlich.

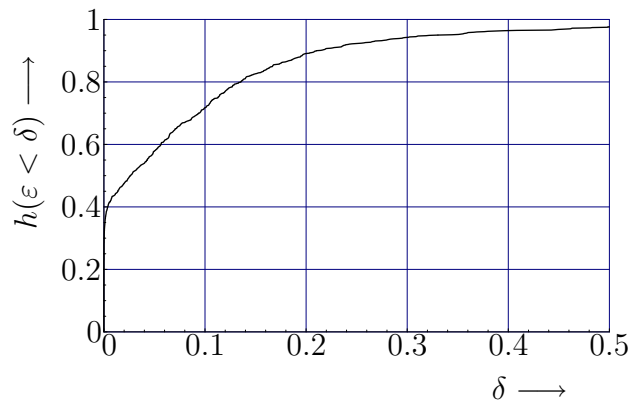
### Quantil der Warteschlangenlänge

Auch die Warteschlangenquantile können sehr gut approximiert werden. Man erkennt aber, dass es für die zufällig gewählten Parameter kaum Warteschlangenquantile größer als 5 gibt (Abbildung 4.13). Wenn die Parameterbereiche an andere Bedürfnisse der Praxis angepasst werden, könnte eine Ungleichverteilung zugunsten des Trainings von Elementen mit hoher Auslastung hier sicherlich Abhilfe schaffen.



**Abbildung 4.13:** Einschleusung, Quantil der Warteschlange auf der Nebensacke. Vergleich der Netzdaten  $Q_N$  mit den Simulationsdaten  $Q_S$

Die Verteilung der Fehler (Abbildung 4.14) zeigt, dass der Fehler  $\varepsilon \ll 0.5$  bleibt und somit nicht ins Gewicht fällt.



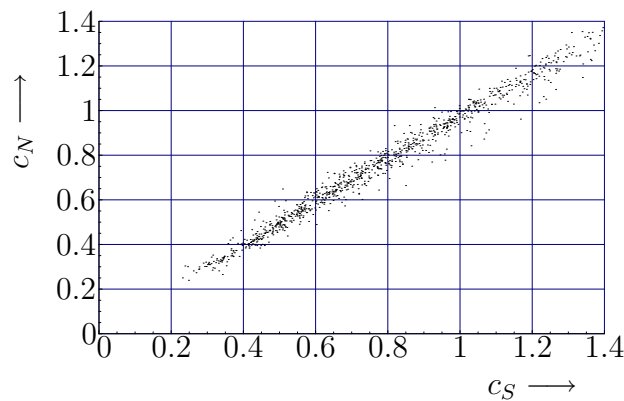
**Abbildung 4.14:** Einschleusung, Quantil der Warteschlange auf der Nebensacke. Verteilung der Netzfehler  $\varepsilon$

## Validierung

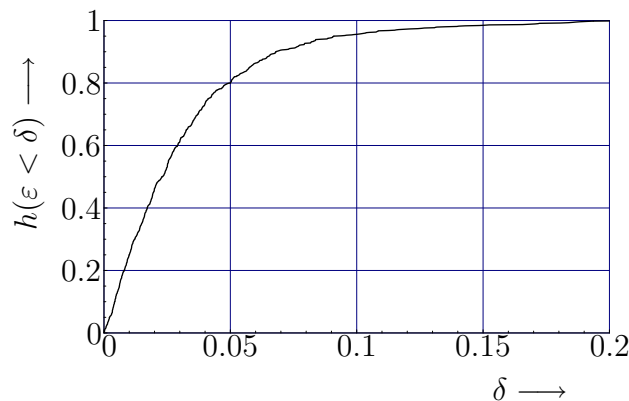
Für die Erzeugung der Trainingsdaten für die neuronalen Netze wurden Transportsystem-Elemente für verschiedene Parameterkonfigurationen mit gammaverteilten Eingangsströmen gespeist, um daraus die interessierenden statistischen Parameter zu gewinnen. Für die Validierung wird jetzt für die Nebenstrecke anstelle der Gammaverteilung eine Zweipunktverteilung benutzt (siehe Vergleich S. 48). Die Ergebnisse der Simulationen werden mit den Netzprognosen für dieselben Parameter verglichen.

### *Variationskoeffizient der Abgangsverteilung*

Abbildung 4.15 zeigt die Abweichungen der Netzdaten von den Referenzwerten. Die zugehörige Verteilung der Fehler ist in Abbildung 4.16 dargestellt.



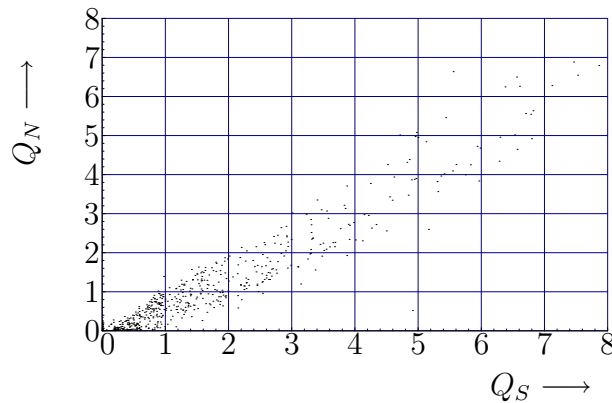
**Abbildung 4.15:** Einschleusung, Variationskoeffizient des Ausgangs (Zweipunktverteilung). Vergleich der Netzdaten  $c_N$  mit den Simulationsdaten  $c_S$



**Abbildung 4.16:** Einschleusung, Variationskoeffizient des Ausgangs (Zweipunktverteilung). Verteilung der Netzfehler  $\varepsilon$

Mit einer Wahrscheinlichkeit von 80% ist die Abweichung kleiner als 0.05, das entspricht einem relativen Fehler von etwa 5%. Dieses Ergebnis ist beachtlich, wenn man sich die Unterschiedlichkeit der Verteilungsfunktionen vor Augen führt!

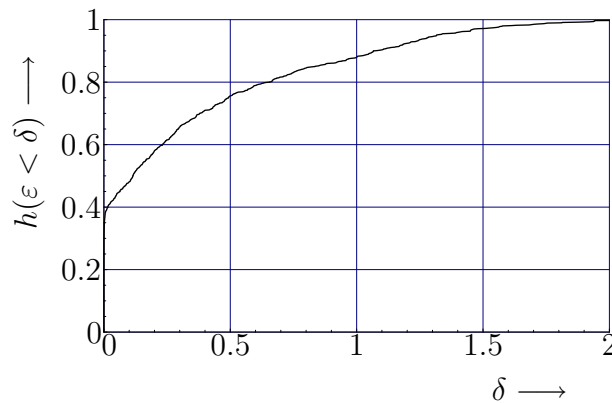
### Quantil der Warteschlangenlänge



**Abbildung 4.17:** Einschleusung, 90%-Quantil der Warteschlange auf der Nebestrecke (Zweipunktverteilung). Vergleich der Netzdaten  $Q_N$  mit den Simulationsdaten  $Q_S$

Wie nicht anders zu erwarten war, weichen die Warteschlangen am zweipunktverteilten Eingang deutlich von denen ab, die dieses Modell prognostiziert (Abbildung 4.17).

Betrachtet man jedoch die Häufigkeitsverteilung des absoluten Fehlerbetrags (Abbildung 4.18), so erkennt man: Mit einer Wahrscheinlichkeit von 90% ist die Abweichung höchstens  $\pm 1$  Stauplatz ( $\varepsilon < 1$ ), mit über 70% stimmen Prognose und Simulation überein ( $\varepsilon < 0.5$ ).



**Abbildung 4.18:** Einschleusung, 90%-Quantil der Warteschlange auf der Nebestrecke (Zweipunktverteilung). Verteilung der Netzfehler  $\varepsilon$



### 4.2.2 Ausschleusung

#### Beschreibung des Transportsystem-Elements

Ein weiteres wichtiges Element in einem Transportsystem ist eine Verzweigung. Hier soll als Beispi­e dafür die folgendermaßen definierte *Ausschleusung* betrachtet werden (Abbildung 4.19): Die Hauptstrecke bewegt sich mit einer konstanten Geschwindigkeit  $v$ . Der Mindestabstand zwischen zwei Transporteinheiten (Vorderkante zu Vorderkante) ist größer als  $s_{min}$  bzw.  $t_{min} = s_{min}/v$ .

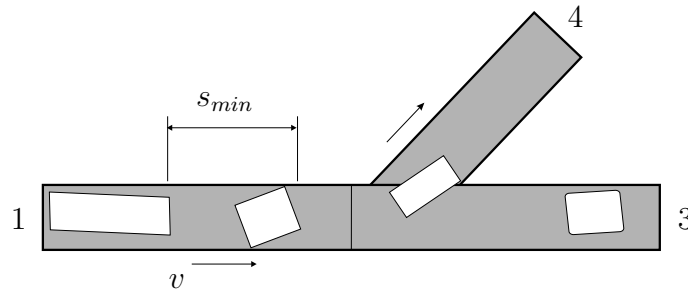


Abbildung 4.19: Ausschleusung

Transporteinheiten vom Eingang 1 können auf der Hauptstrecke in Richtung 3 befördert werden oder mit einer Transferzeit  $t_{tr}$  auf die Nebenstrecke 4 ausgeschleust werden. Für die Dauer der Passage der Ausschleusung  $t_{min}$  bzw.  $t_{min} + t_{tr}$  kann keine weitere Transporteinheit das Element benutzen.

#### Modelldaten

Analog zur Einschleusung (Abschnitt 4.2.1) werden die folgenden Daten benutzt:

- $t_{min} \dots$  Mindestzeitabstand (Vorderkante zu Vorderkante) zwischen aufeinander folgenden Transporteinheiten (Hauptstrecke),
- $t_{tr} \dots$  Zeit für den Transfer<sup>13</sup> einer Transporteinheit von der Hauptstrecke auf die Nebenstrecke,
- $EX \dots$  mittlerer Abstand zwischen den Ankunftszeiten aufeinander folgender Transporteinheiten auf der Hauptstrecke,
- $c_X \dots$  Variationskoeffizient der Ankunftszeiten der Hauptstrecke.
- $a \dots$  ( $=a \cdot 100\%$ ) Anteil der geradeaus fahrenden Transporteinheiten. Der Anteil der ausgeschleusten Transporteinheiten ist  $1 - a$ .

#### Parameterraum für die Simulationen

Auch hier soll die Auslastung  $\rho$  für die Parameterauswahl benutzt werden. Wenn  $n = \frac{T}{EX}$  die Anzahl der in der Zeit  $T$  eintreffenden Transporteinheiten ist, errechnet

<sup>13</sup>Bei rechtwinkligen Ausschleusungen ist die Transferzeit  $t_{tr} > 0$ , für schräge Ausschleusungen (Kamm-Pusher, Kettenausschleuser mit Mitnehmer) kann  $t_{tr}$  auch Null sein.

sich die Zeit  $T_G$ , die für die Bearbeitung dieser Transportaufträge benötigt wird, wie folgt:

$$T_G = n \cdot a \cdot t_{min} + n(1 - a)(t_{min} + t_{tr}). \quad (4.18)$$

Aus  $\varrho = T_G/T$  folgt:

$$EX = \varrho \cdot (a \cdot t_{min} + (1 - a)(t_{min} + t_{tr})). \quad (4.19)$$

Wie auch bei der Einschleusung werden sämtliche Zeiten auf  $t_{min}$  normiert, damit kann dieser Parameter wegfallen. Als sinnvolle Bereiche werden  $0 < a < 1$ ,  $0 \leq t_{tr} \leq 2$ ,  $0 < \varrho < 1$  und  $0.2 \leq c_X \leq 1.5$  angesehen. Damit kann die Simulation über diesem Parameterbereich für 4000 Parametertupel (3000 zum Training, 1000 zur Validierung) durchgeführt werden.

#### Parameter für das Netztraining

Für das Training der neuronalen Netze werden die Parameter transformiert. Es werden  $\frac{1}{EX}$ ,  $a$ ,  $t_{tr}$  und  $c_X$  als Eingangsdaten für die Netze benutzt. Dabei entspricht  $\frac{1}{EX}$  der mittlere Ankunftsrate.

Es wurden für die drei zu approximierenden Parameter Netze gleicher Topologie benutzt (Abbildung 4.20):

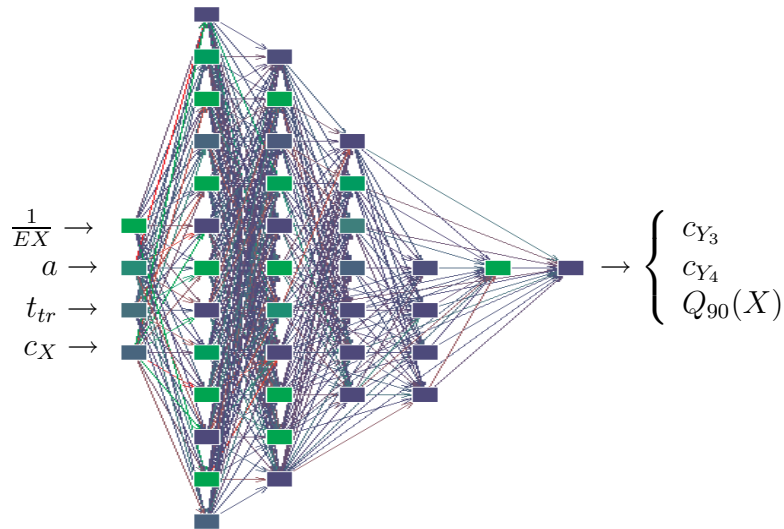


Abbildung 4.20: Ausschleusung. Topologie der neuronalen Netze

### Verifikation und Validierung

Die neuronalen Netze konnten erfolgreich trainiert werden. Die Approximationen der einzelnen Parameter sind ähnlich gut wie bei der Einschleusung. Aus diesem Grunde wird auf die detaillierte Diskussion an dieser Stelle verzichtet.

Für die Validierung wurde vorausgesetzt, dass der Eingangsstrom einer Zweipunktverteilung folgt. Die entsprechenden Diagramme zur Verifikation und Validierung finden sich im Anhang B.1 auf S. 97.

Als Ergebnis der Validierungen soll hier nur erwähnt werden, dass für die Hauptstrecke mit einer Wahrscheinlichkeit von 80% die Fehler für die Variationskoeffizienten unter 0.1 und für das Warteschlangenquantil des Eingangs unter 1 liegen.

### 4.2.3 Kreuzung

#### Beschreibung des Transportsystem-Elements

Eine *Kreuzung* ist in diesem Modell ein Transportsystem-Element mit zwei Eingängen und zwei Ausgängen (siehe Abbildung 4.21). Dabei wird die Strecke von 1 nach 3 als Hauptstrecke und die von 2 nach 4 als Nebenstrecke bezeichnet. Die Kreuzung besitzt eine Transportvorrichtung (etwa ein Hubwerk oder eine Drehvorrichtung), welche drei Zustände hat:

Zustand A: Es können Transporteinheiten von Richtung 1 aufgenommen oder nach Richtung 3 abgegeben werden.

Zustand B: Es können Transporteinheiten von Richtung 2 aufgenommen oder nach Richtung 4 abgegeben werden.

Zustand C: Die Transportvorrichtung bewegt sich zwischen den beiden Zuständen. Es können weder Transporteinheiten aufgenommen noch abgegeben werden.

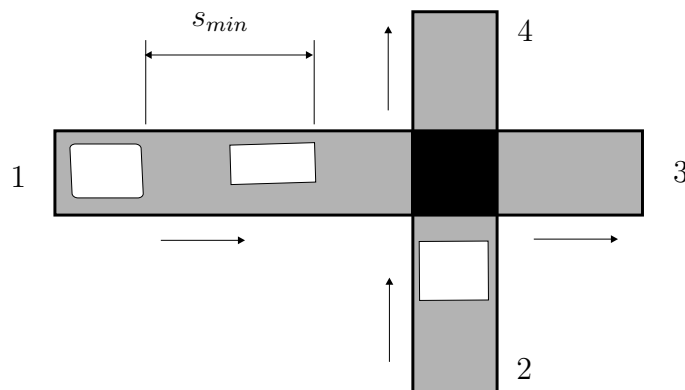


Abbildung 4.21: Kreuzung

Die Steuerung für eine Kreuzung lässt sich etwa folgendermaßen beschreiben:

1. Wähle aus, welcher von beiden Eingängen als nächster bedient wird.
2. Wenn nötig, positioniere Transportvorrichtung (Positionierzeit  $t_m$ ).
3. Übernimm Transporteinheit auf die Vorrichtung (Transferzeit  $t_{tr}$ ).
4. Wenn nötig, positioniere Transportvorrichtung (Positionierzeit  $t_m$ ).
5. Gib Transporteinheit an den entsprechenden Ausgang ab (Transferzeit  $t_{tr}$ ).

Über die „Transportvorrichtung“ für den Transfer zwischen Ein- und Ausgängen wird nichts weiter ausgesagt. Damit können beispielsweise Hub- oder Drehtische mit zwei Ein- und Ausgängen als „Kreuzung“ im obigen Sinne angesehen werden.

Wesentlich für das Transportsystem-Element ist, dass für die Bewegung zwischen den beiden möglichen Positionen eine Zeit  $t_m > 0$  und für die Lastübernahme bzw. Lastübergabe eine Zeit  $t_{tr} > 0$  benötigt wird.

Für die Auswahl des jeweils nächsten Transportauftrages ist eine Steuerung zuständig. Entsprechend der Modellannahmen muss diese Steuerung lokal sein. Für den Rahmen dieser Arbeit werden zwei Standardprinzipien ausgewählt:

- a) Die Regel *first-come-first-serve* bedient jeweils die zuerst anfragende Transporteinheit (FCFS). In dieser Modellklasse wird nicht berücksichtigt, wie lange die Transporteinheit schon in einer Warteschlange *vor* der Kreuzung wartet.
- b) Eine einfache Vorfahrtsregel sorgt dafür, dass die Transporteinheiten der Hauptstrecke (Eingang 1) bevorzugt bedient werden.

Sowohl für die Auswahl des nächsten Auftrages als auch für die Festlegung einer Parkposition der Transportvorrichtung für die inaktive Zeit gibt es reichlich Spielraum. Für jede dieser Steuerungen sind separate Simulationen und Training der neuronalen Netze nötig. Um den Umfang dieser Arbeit nicht zu sprengen, beschränkt sie sich auf die beiden vorgestellten Steuerungen, bei denen die Transportvorrichtung nach beendetem Transport an Ort und Stelle auf den nächsten Auftrag wartet.

## Modelldaten

Im Unterschied zu Ein- und Ausschleusung kann die Auslastung  $\varrho$  hier nicht berechnet und als Parameter für die Simulation genutzt werden. Grund dafür ist, dass die notwendigen Steuerungsregeln die analytische Handhabbarkeit verhindern. Stattdessen ist  $\varrho$  auch eine durch das Modell zu ermittelnde Größe.

Um aber trotzdem eine Abschätzung für  $\varrho$  treffen zu können, wird die lokale Transportmatrix benutzt:

$$M = \begin{pmatrix} 0 & 0 & n_{13} & n_{14} \\ 0 & 0 & n_{23} & n_{24} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (4.20)$$

Dabei sind  $n_{ij}$  die mittleren Anzahlen der Transporte von  $X_i$  nach  $Y_j$  in einer festen Zeit  $T$  ( $i = 1, 2, j = 3, 4$ ). Sei  $n = \sum n_{ij}$  die Summe aller Bewegungen. Hier interessiert eine Abschätzung der gesamten Zeit  $T_G$ , die für diese Bewegungen benötigt wird. Sie besteht aus den Teilzeiten  $T_G = T_m + t_m$ , dabei ist  $T_m$  die Zeit für die Transfers und  $t_m$  die Zeit, die die leeren Positionierungen kosten. Offenbar ist

$$T_m = n_{13}(2 \cdot t_{tr}) + n_{14}(2 \cdot t_{tr} + t_m) + n_{23}(2 \cdot t_{tr} + t_m) + n_{24}(2 \cdot t_{tr}) \quad (4.21)$$

$$= 2t_{tr}(n_{13} + n_{24}) + (2 \cdot t_{tr} + t_m)(n_{14} + n_{23}). \quad (4.22)$$

Für  $t_m$  hilft nur eine Abschätzung: Vorausgesetzt sei, dass beide Eingangsströme unabhängig sind. Seien mit  $p_{ij} = n_{ij}/n$  die Wahrscheinlichkeiten für eine Bewegung

von  $X_i$  nach  $Y_j$  bezeichnet. Dann ist  $p_{13} + p_{23}$  die Wahrscheinlichkeit dafür, dass eine Transporteinheit zum Ausgang 3 befördert und  $(p_{13} + p_{23}) \cdot (p_{23} + p_{24})$  die Wahrscheinlichkeit dafür, dass eine (leere) Positionierung vom Ausgang 3 zum Eingang 2 ausgeführt wird.

Die hierbei erhaltene Positionierzeit  $\widetilde{t}_m$  ist lediglich eine Näherung<sup>14</sup> der tatsächlich benötigten Zeit:

$$\widetilde{t}_m = n \cdot t_m \cdot ((p_{13} + p_{23}) \cdot (p_{23} + p_{24}) + (p_{14} + p_{24}) \cdot (p_{13} + p_{14})). \quad (4.23)$$

Benutzen man  $n_{ij} = n \cdot p_{ij}$ , so erhält man mit (4.22) eine Abschätzung  $\widetilde{T}_G$  für die benötigte Gesamtzeit:

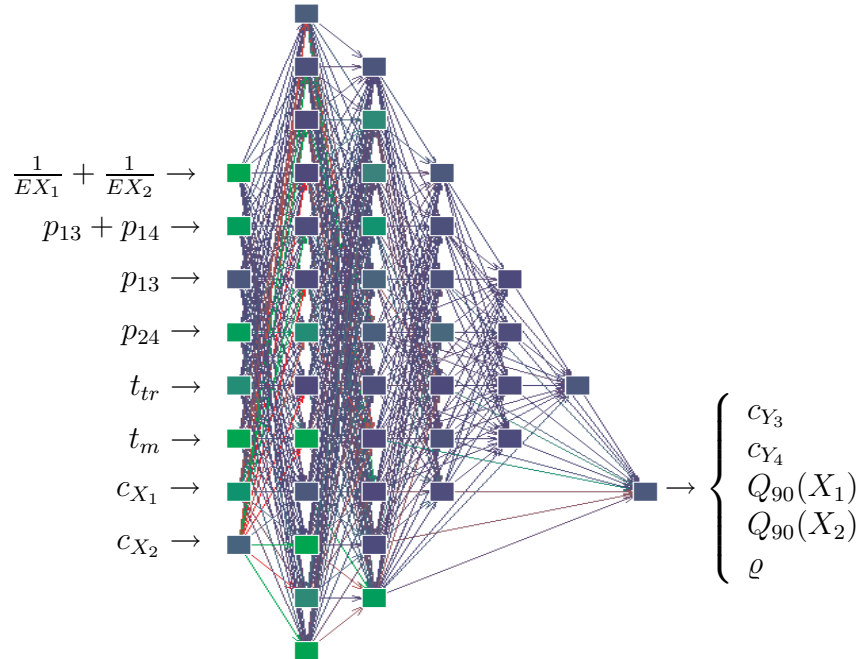
$$\begin{aligned} \widetilde{T}_G = n \cdot (2t_{tr}(p_{13} + p_{24}) + (2 \cdot t_{tr} + t_m)(p_{14} + p_{23}) + \\ t_m \cdot ((p_{13} + p_{23}) \cdot (p_{23} + p_{24}) + (p_{14} + p_{24}) \cdot (p_{13} + p_{14}))). \end{aligned} \quad (4.24)$$

#### Parameter für die Simulation

Man setze wieder voraus, dass alle Ströme das gleiche Minimum  $t_{min}$  besitzen und normiere entsprechend. Außerdem sei  $0.5 \leq t_{tr} \leq 2.5$ ,  $0.5 \leq t_m \leq 2.5$ . Die Variationskoeffizienten liegen zwischen 0.2 und 1.5.

Als freie Parameter werden  $0 < \varrho < 1$  und die Transferwahrscheinlichkeiten  $p_{ij} > 0$  mit  $\sum p_{ij} = 1$  gewählt. Mit  $\varrho = \widetilde{T}_G/T$  und Gleichung (4.24) kann dann einen Wert für  $n$  berechnet und damit die Transportmatrix (4.20) bestimmt werden.

Für beide Steuerungen werden separate Simulationen durchgeführt. Man erhält einen doppelten Satz von Trainings- und Testdaten, die auch unabhängig voneinander ausgewertet werden.



**Abbildung 4.22:** Kreuzung. Topologie der neuronalen Netze

<sup>14</sup>sie ist korrekt für eine *first-come-fist-serve*-Steuerung

### Parameter für das Netztraining

Für das Training der Netze werden die Daten so transformiert, dass man die Eingangsdaten der neuronalen Netze entsprechend Abbildung 4.22 erhält. Wie bereits in Abschnitt 4.1.5 erwähnt, wird für jede Zielgröße (und für jede Steuerung) ein separates Netz trainiert. In jedem Fall scheint die angegebene Topologie ausreichend zu sein.

## Verifikation und Validierung

Obwohl sich die Verteilungen der Fehler für beide Steuerungen unterscheiden, sollen sie hier kurz gemeinsam diskutiert werden. (Alle Diagramme sind im Anhang B hinterlegt. Die Darstellungen für Kreuzungen mit FCFS-Steuerung finden sich auf S. 98, für Vorfahrtsregelung siehe S. 99.)

Für die Validierung wird die Hauptstrecke mit einem gammaverteilten, die Nebenstrecke mit einem zweipunktverteilten Eingangsstrom gespeist. Selbst in diesem Fall können die Variationskoeffizienten der Abgangsströme mit einer Wahrscheinlichkeit von 80% bis auf einen Fehler von unter 0.07 genähert werden.

Die Prognosen für die Warteschlangen der Hauptstrecke stimmen zu 90% mit den Simulationsdaten überein ( $\varepsilon < 0.5$ ). Auf der Nebenstrecke, deren Eingangsstrom zweipunktverteilt ist, beträgt der Fehler für 80% der beobachteten Fälle weniger als  $\pm 1$  Stauplatz.

### 4.2.4 Regalbediengerät

#### Beschreibung des Transportsystem-Elements

Regalbediengeräte (RBG) sind Fördermittel, mit denen auf Transporteinheiten in einem Warenlager zugegriffen werden kann [Jün00]. Im weiteren Sinne können das sowohl automatische als auch manuelle Vorrichtungen sein. In besonderen Fällen werden die Regalbediengeräte so ausgestattet, dass sie die Regalgasse wechseln können. Solch ein Umsetzen ist dann sinnvoll, wenn die Anzahl der Ein- und Auslagerungsaufträge für eine Gasse deutlich unter der Leistungsgrenze des Regalbediengerätes liegt. Im Normalfall sind Regalbediengeräte jedoch an eine Regalgasse gebunden. Auf diese beschränkt sich die vorliegende Arbeit.

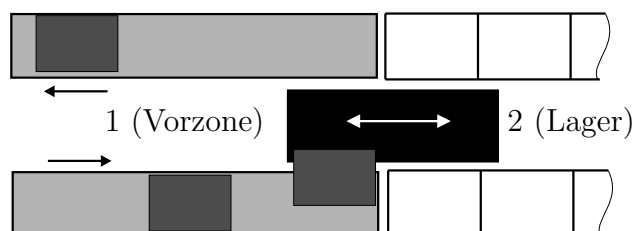


Abbildung 4.23: Regalbediengerät bei Lastübernahme (Einlagerung)

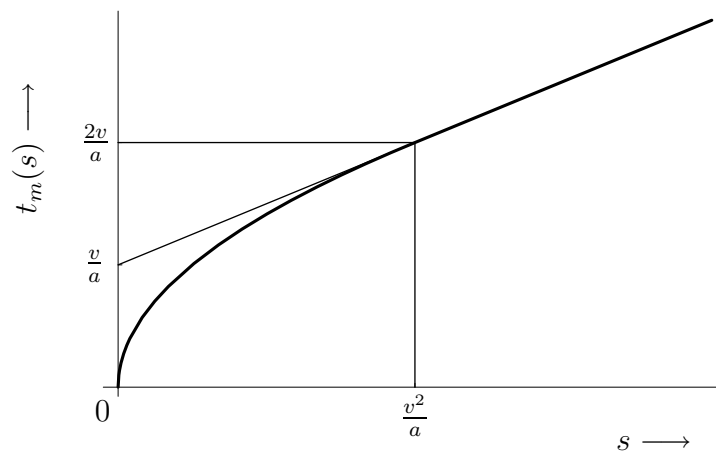
Charakteristische Größen für ein Regalbediengerät in diesem Modell sind die folgenden:

- Länge  $L$  und Höhe  $H$  des Regals,
- Fahrgeschwindigkeiten in horizontaler ( $v_h$ ) und vertikaler ( $v_v$ ) Richtung,
- Eine feste Zeit  $t_{tr}$  für die Lastübernahme bzw. -übergabe.
- Eine konstante Zeit  $t_p$  für die Feinpositionierung.

Obwohl in der Praxis diskrete Größen, wie die Anzahl von Fächern in horizontaler und vertikaler Richtung eine Rolle spielen, werden sie bei diesem Modell nicht berücksichtigt. Um das Modell klein zu halten, werden die folgenden Festlegungen getroffen:

- Der Ein- und Auslagerungspunkt liegt an einer Ecke der Regalfläche. Im allgemeinen Fall müßten ein bzw. zwei Parameter zusätzlich in das Modell aufgenommen werden.
- Es wird auf jedwede Sortierung der Lagereinheiten verzichtet. Wie bei der Modellbildung (S. 37) erwähnt wird, ist für jede Steuerung ein eigenes Modell erforderlich. Für eine genauere Modellierung sind darüberhinaus Daten über Zugriffshäufigkeiten etc. erforderlich, was den Rahmen dieses Modells sprengen würde.
- Wird das RBG nicht benötigt, bleibt es an Ort und Stelle stehen.<sup>15</sup> Hier gilt das Gleiche, wie auch bei den hier beschriebenen Kreuzungen: jede veränderte Steuerung erfordert ein eigenes Modell.
- Das Lastaufnahmemittel ist nur in der Lage, eine Lagereinheit zu befördern. Bei Regalen mit Mehrfach-Lastaufnahmemitteln wären intensive Überlegungen zur Routenplanung nötig [Kre01].

Für die beschleunigte Bewegung des Regalbediengerätes gilt folgendes:



**Abbildung 4.24:** Zeitbedarf  $t_m$  für eine Entfernung  $s$  (in einer Richtung) und lineare Näherung  $t_m(s)$

<sup>15</sup>In den Simulationsläufen werden die Ein- und Auslagerungsraten variiert. Eine Strategie, die Einlagerungen bevorzugt - etwa das Parken an der Einlagerungsposition - ist bei erhöhten Auslagerungsraten ungünstig.

Die Zeit  $t_m$ , die für eine komplette Bewegung in einer Richtung (inklusive Beschleunigung und Abbremsen mit  $a$  bzw.  $-a$ ) für eine Entfernung  $s$  benötigt wird, ist

$$t_m(s) = \begin{cases} 2\sqrt{s/a} & \text{für } s \leq v^2/a, \\ v/a + s/v & \text{sonst} \end{cases}. \quad (4.25)$$

Abbildung 4.24 zeigt, in welchem Bereich die Näherung von  $t_m(s)$  durch  $\widetilde{t}_m(s) := \frac{v}{a} + \frac{s}{v}$  zu Abweichungen führt. Insbesondere dann, wenn das Lager gut sortiert ist, sind die zu bewältigenden Entfernungen klein. Doch dieser Fall wurde bereits aus anderen Gründen ausgeschlossen.

Der Parameter für die Beschleunigung kann somit in erster guter Näherung eingespart werden, indem die Zeit  $v/a$  zu der eigentlichen Positionierzeit  $t_p$  addiert wird.

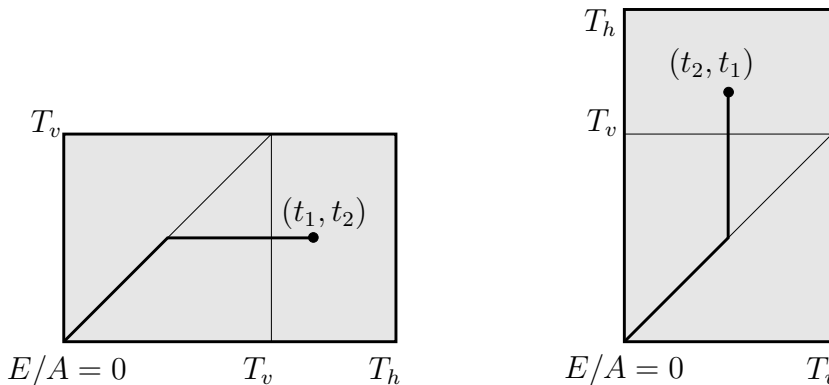
### Modelldaten

Gegeben sei der (Daten-)Strom der Auslagerungen  $(0, EX_2, c_{X_2})$ . Der minimale Abstand zweier Forderungen ist 0, weil im realen System über das Datennetz Auslagerungsaufträge aus dem ganzen System unabhängig voneinander eingehen können.<sup>16</sup>

#### Parameter für die Simulation

Um aus den gegebenen Parametern ein längeninvariantes Modell abzuleiten, geht man von Höhe  $H$  und Hubgeschwindigkeit  $v_v$  über zu  $T_v = H/v_v$ , der maximalen vertikalen Fahrzeit. Entsprechend ist  $T_h = B/v_h$  die maximale horizontale Fahrzeit). Auf diese Weise kann für ein Lagerpunkt über die Zeiten der Einzelbewegungen identifiziert werden, die vom Ein- und Auslagerungspunkt „0“ benötigt werden.

Die Fahrzeit zwischen zwei Punkten der Lagerfläche wird jetzt als Maß für die Entfernung der Punkte benutzt. Da beide Fahrwerke auch simultan betrieben werden können, ergibt sie sich aus dem Maximum der benötigten einzelnen Fahrzeiten (Maximum-Metrik).



**Abbildung 4.25:** Maximum-Metrik gestattet Vertauschen der Fahrzeiten

<sup>16</sup>Der tatsächlich vorhandene Abstand, der aus der Signalverarbeitung (Lagerverwaltungsrechner) resultiert, liegt im Bereich von Sekundenbruchteilen und soll deshalb hier nicht interessieren.



Das Lager mit den Fahrzeiten  $T_h$  und  $T_v$  ist offensichtlich nicht mehr von dem „gedrehtem“ Lager mit der horizontalen Fahrzeit  $T_v$  und der vertikalen Fahrzeit  $T_h$  unterscheidbar (Abbildung 4.25).

Für die Simulation wird der Regalwandparameter  $b = \frac{T_v}{T_h}$  benutzt. In der Praxis sind für ihn Werte zwischen 0.7 und 2 gebräuchlich. Für die Modellierung können  $T_h$  und  $T_v$  vertauscht werden, ohne dass sich die Ergebnisse verändern. Somit kann man sich auf Parameter  $0.5 \leq b \leq 1$  beschränken.

$X_1$  sei der Strom der einzulagernden Transporteinheiten,  $X_2$  der Datenstrom der Auslagerungsaufträge.  $Y_1$  sei der Ausgangsstrom des Lagers,  $Y_2$  der Datenstrom der Einlagerungsbestätigungen.

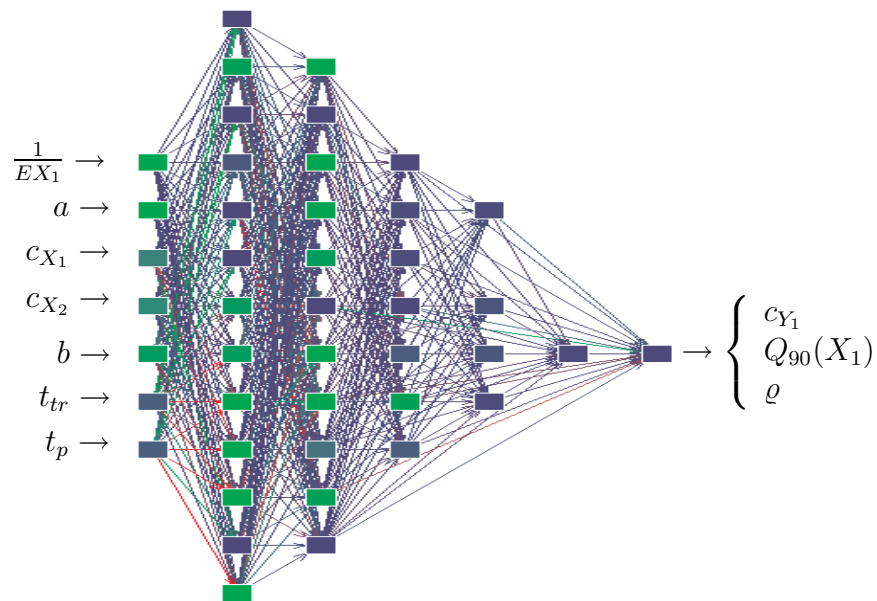
In diesem Modell werden die Warteschlange der Auslagerungsaufträge ( $X_2$ ) und der Variationskoeffizient des Datenstroms der Einlagerungsbestätigungen  $c_{Y_2}$  nicht näher betrachtet. Die Parameter der Stromes  $X_2$  werden jedoch für die Modellierung benutzt.

Der Anteil von Einlagerungen an den gesamten Aufträgen wird dann mit Hilfe des Parameters  $a = \frac{EX_1}{EX_1+EX_2}$  beschrieben ( $0 < a < 1$ ).

Da die Bedienzeit des Gerätes ein Mehrfaches des minimalen Abstandes  $t_{min}$  des Eingangsstroms ist, spielt jene Größe kaum eine Rolle. Wird dieser Wert festgehalten, kann damit die Dimension des Problems um 1 verringert werden. Alle Zeiten der Ströme werden auf  $T_h$  normiert und damit die Dimension wiederum um 1 verkleinert.

#### Parameter für das Netztraining

Die Eingangsdaten und die Topologie für die neuronalen Netze werden in Abbildung 4.26 dargestellt.



**Abbildung 4.26:** Regalbediengerät. Topologie der neuronalen Netze

## Verifikation und Validierung

Zur Validierung werden für die Auslagerung gammaverteilte und für die Einlagerung zweipunktverteilte Zwischenzeiten der jeweiligen Ereignisse der Ströme vorausgesetzt.

Die erfolgreichen Trainings- und Testläufe werden mit den Diagrammen im Anhang B auf den Seiten 101 f. dokumentiert. Zusammenfassend können für eine 80%ige Wahrscheinlichkeit bei zweipunktverteilten Eingangsströmen die folgenden Fehlerabschätzungen für die Prognosen der neuronalen Netze getroffen werden:

Der Variationskoeffizient des Ausgangs weicht um weniger als 0.02, das Warteschlangenquantil um weniger als 1, die Auslastung um weniger als 0.01 von den Simulationen ab.

### 4.2.5 Bedienstation

#### Beschreibung des Transportsystem-Elements

Eine *Bedienstation* hat im Rahmen dieses Modells die folgenden Eigenschaften:

- Sie hat genau einen Eingang und einen Ausgang.
- Es können nicht mehrere Transporteinheiten gleichzeitig bedient werden. (Einzelabfertigung)
- Die Bedienzeit ist gammaverteilt.

Elemente eines Transportsystems, wie etwa Verpackungsmaschinen, Kontrollstationen oder Scanner usw., können als Bedienstation mit gammav verteilter Bedienzeitverteilung recht gut modelliert werden.

Wie bei den anderen Modellklassen ist der Mindestabstand des Eingangsstroms  $t_{min} = s_{min}/v > 0$ , dabei ist  $v$  die Transportgeschwindigkeit und  $s_{min}$  der minimale Abstand (Vorderkante zu Vorderkante zweier aufeinanderfolgender Transporteinheiten (Abbildung 4.27).

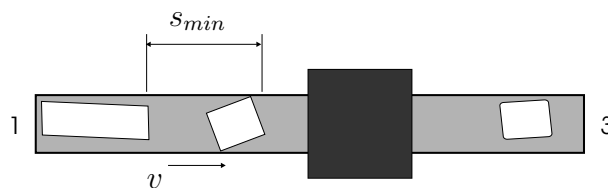


Abbildung 4.27: Bedienstation

#### Modelldaten

Die Bedienstation hat demzufolge auch nur zwei Parameter, nämlich den Erwartungswert  $EB$  und den Variationkoeffizienten  $c_B$  der Bedienzeitverteilung.

#### Parameter für die Simulation

Für den Erwartungswert der Eingangsverteilung werden Werte zwischen  $t_{min}$  und  $10 \cdot t_{min}$  zur Simulation zugelassen. Die Auslastung  $\varrho = EX/EB$  mit  $0.2 \leq \varrho < 1$  legt

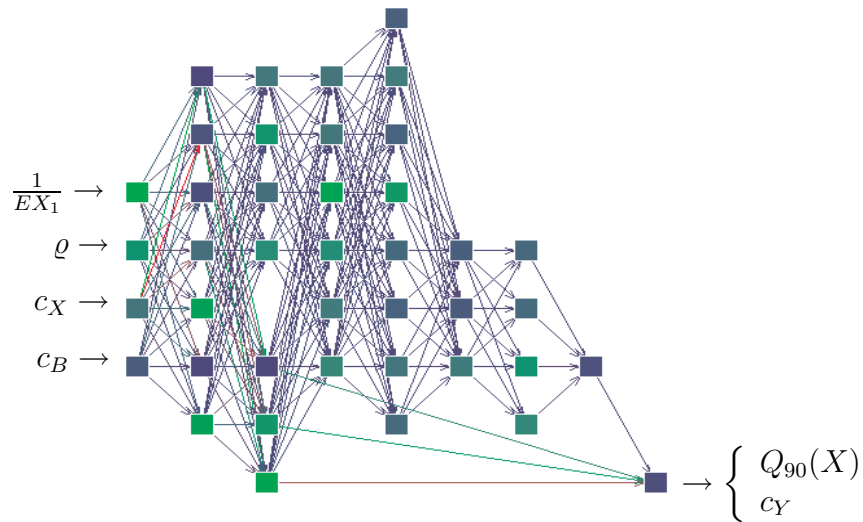
implizit die mittlere Bedienzeit  $EB$  fest. Wie auch bei den anderen Modellklassen, normiert man auf  $t_{min}$ . Sowohl  $c_X$  als auch  $c_B$  nehmen Werte zwischen 0.2 und 1.5 an.

Wegen der geringen Dimension der Aufgabe reichen hier schon etwa 2000 Simulationen aus, ohne ein Übertrainieren zu riskieren. Der Umfang der Verifikationsdaten wird mit 1000 Datensätzen beibehalten.

#### *Parameter für die Simulation*

Die folgenden Eingangsdaten werden für das Training genutzt:  $1/EX$  (entspricht der mittleren Ankunftsrate),  $\varrho$ ,  $c_X, c_B$ .

Die neuronalen Netze für die Approximation des Warteschlangenquantils bzw. des Variationskoeffizienten der Abgangsverteilung haben die gleiche Topologie<sup>17</sup> (Abbildung 4.28).



**Abbildung 4.28:** Bedienstation. Topologie der neuronalen Netze

## Verifikation und Validierung

Wie die Abbildungen im Anhang B ( S. 102 f.) deutlich zeigen, konnten die neuronalen Netze erfolgreich trainiert werden.

Die Validierungen mit zweipunktverteilten Eingangsströmen zeigen, wie erwartet, deutliche Abweichungen. Mit einer Wahrscheinlichkeit von knapp 80% weicht die Vorhersage des Variationskoeffizienten der Abgangsverteilung weniger als 0.15 von den Referenzdaten ab. Um hierfür das Vertrauen in das Modell zu stärken, könnten weitere Versuche der Validierung (z.B. mit praxisnahen Eingangsverteilungen) unternommen werden. Jedoch kann das Warteschlangenquantil mit einer 90%-igen Wahrscheinlichkeit auf plus/minus einen Stauplatz genau prognostiziert werden.

<sup>17</sup>Die Optimierung der Netzstruktur ist nicht Kern dieser Arbeit. Es ging primär um Machbarkeitsstudien, so dass oftmals eine Netztopologie wieder benutzt wurde, wenn sie sich bereits an anderer Stelle bewährt hat. Eine günstige Struktur für die Netze zu finden, wäre indes Mittel zum Zweck, die Trainingszeit zu verkürzen.

## 4.3 Zusammenfassung

Für sämtliche untersuchten Modelle können die folgenden Aussagen getroffen werden:

- Die Variationskoeffizienten der Abgangsströme werden für gammaverteilte Eingangsströme exakt vorhergesagt. Für die betrachteten Eingangsströme mit Zweipunktverteilung ist die Genauigkeit auch noch sehr gut.
- Die Auslastung der Transportsystem-Elemente kann mit dem Modell mit hoher Genauigkeit für beide Eingangsverteilungen genähert werden.
- Die Quantile der Warteschlangen können für gammaverteilte Eingangsströme sicher angegeben werden. Für zweipunktverteilte Eingangsströme hängt die Güte der Näherung vom Typ des Elements und von der Länge der Warteschlange ab. Generell liegt die Prognose mit hoher Wahrscheinlichkeit im Bereich  $\pm 1$  Stauplatz.

Durch die Validierung der einzelnen Modellklassen konnte gezeigt werden, dass der Prognosefehler der numerischen Modellierung sehr klein ist. Aus diesem Grunde ist es gerechtfertigt, diese (Teil-)Modelle zu einer komplexeren Struktur zu verknüpfen. Dieses hybride Modell wird im nächsten Kapitel beschrieben.

# 5 Hybride Modellierung

Im vorangegangenen Kapitel wurde ein Modell zur Prognose statistischer Kenngrößen eines Transportsystem-Elements geschaffen und für verschiedene grundlegende Modellklassen implementiert und getestet. Bei allen hier vorgestellten Modellklassen ist eine Verknüpfung zu einem zusammengesetzten Modell sinnvoll.

Mit der Vorlage eines komplexen Modells können einerseits die einzelnen Elemente miteinander logistisch verbunden werden. Auf diese Weise ist es möglich, größere Strukturen auf einfache Weise abzubilden und Aussagen über das Gesamtsystem zu erhalten. Auf der anderen Seite können damit die Modellklassen durch den „Einsatz in der Praxis“ validiert werden.

Im folgenden Abschnitt wird dieses hybride Modell vorgestellt. Es folgen Beispiele, bei denen seine Prognosen mit den Ergebnissen diskreter Simulationen verglichen werden.

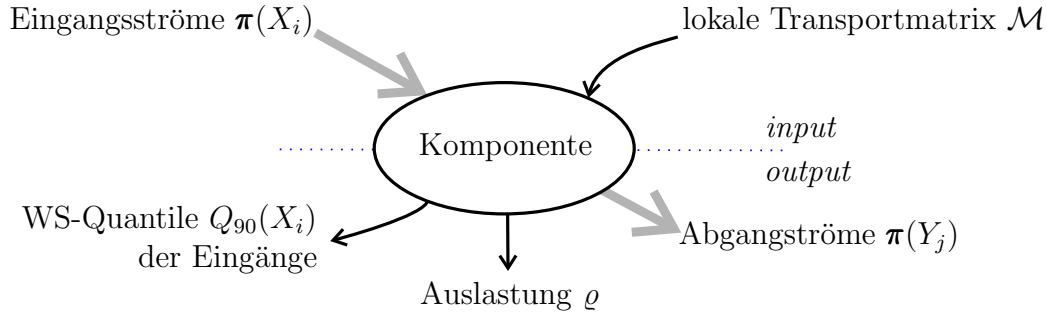
## 5.1 Aufbau eines hybriden Modells

Kern dieses Modells ist Zusammensetzung (Komposition) verschiedener statistischer Teilmodelle zu einem komplexem System. Auf diese Weise können ganze Materialfluss-Systeme oder interessierende Teilbereiche abgebildet werden.

Allerdings gelten auch hier zum Teil beträchtliche Einschränkungen (vgl. auch Abschnitt 4.1, S. 32 ff.). Darum ist das vorgestellte Modell als das zu betrachten, was es ist: eine schnelle Prognose auf Basis statistischer Daten.

**Bemerkung:** Es ist daher nötig, deutlich zu unterscheiden zwischen einem realen Materialfluss-System und dem hier vorgestellten Modell des Systems. Wenn ein Planer ein Transportsystem ohne Blockierfreiheit mit diesem Modell abbildet, muss er sich darüber im Klaren sein, dass wegen der Verletzung der Modellannahmen die dabei berechneten Ergebnisse keine direkte Entsprechung im Originalsystem mehr haben (siehe S. 9). Wie dann zum Beispiel eine Aussage, wie „Vor einem RBG stauen sich mit einer Wahrscheinlichkeit von 90% weniger als 3.5 Transporteinheiten.“ für ein System mit zwei Stauplätzen in der Lagervorzone genutzt werden kann, hängt immer vom konkreten Fall ab.

Die Schnittstellen der einzelnen Bausteine des hybriden Modells entsprechen denen des im vorangegangenen Kapitel beschriebenen Modells (siehe Abbildung 5.1). Das ist nicht überraschend, da die Modelle für Transportsystem-Elemente im Wesentlichen mit Hinblick auf ihre spätere Vernetzung gestaltet wurden. Es ist in diesem Hybridmodell jedoch nicht notwendig, dass seine einzelnen Komponenten elementarer Natur sind. Sie können jeweils auch komplexe Teilsysteme abbilden.



**Abbildung 5.1:** Schnittstellen einer Komponente des hybriden Modells

Aus den Parametern der Eingangsströme  $\pi(X_i)$  und der lokalen Transportmatrix  $\mathcal{M}$  liefert jede Komponente ihre Prognose für die Abgangströme  $\pi(Y_j)$  - in derselben Parameterdarstellung  $\pi(\cdot)$ . Das ist die Voraussetzung dafür, dass überhaupt eine übergeordnete Struktur aufgebaut werden kann.

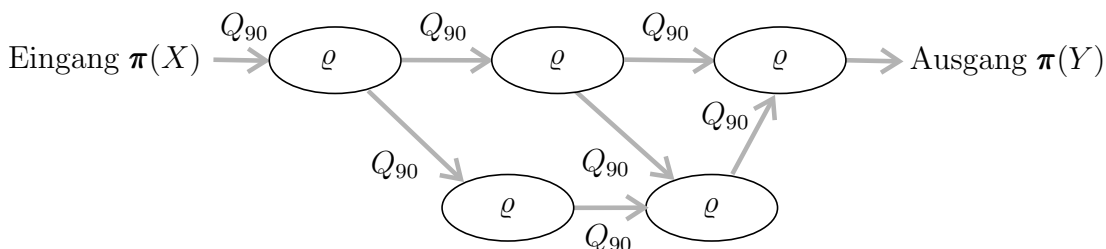
Außerdem, denn das Modell ist nicht Selbstzweck, werden die den Nutzer eigentlich interessierenden Größen berechnet.

Im Rahmen der vorliegenden Arbeit wurden als Parameter der Ströme das Minimum, der Erwartungswert und der Variationskoeffizient der statistischen Verteilungen der Zeiten zwischen zwei Ereignissen des Stroms benutzt (siehe S. 38). Als wichtig für den Betreiber werden die Auslastung  $\varrho$  und die 90%-Quantile der Warteschlangen an den Eingängen  $Q_{90}$  einer Komponente erachtet.

Der konkrete Aufbau der einzelnen Modelle für Transportsystem-Elemente oder Teilsysteme ist aus dieser Sicht irrelevant. Hier sind mehrere Methoden gleichzeitig nebeneinander zugelassen. Die Prognosen können mit Hilfe von neuronalen Netze oder anderen numerischen Ansätzen (z.B. Wavelets [Ant99], [Gra95]) berechnet oder analytisch bestimmt werden. Allen gemeinsam muss lediglich die Schnittstelle (Abbildung 5.1) sein.

Damit wird auch das Attribut „hybrid“ deutlich als Integration verschiedener Modelle und Modellansätze in ein komposites Modell höherer Komplexität (Abbildung 5.2).

Anstelle einer konkreten Routenplanung für die Transporteinheiten treten in jedem Teilmodell die lokalen Transportmatrizen, die über die Häufigkeiten von Transporten innerhalb der modellierten Elemente bzw. Teilsysteme Auskunft geben. Damit werden die mittleren Transportraten im hybriden Modell strukturtreu abgebildet.



**Abbildung 5.2:** Verknüpfung der Komponenten

### 5.1.1 Berechnung der Parameter

Voraussetzung für die Berechnung der interessierenden Werte für die Auslastungen der Elemente bzw. Teilsysteme und Quantile der Warteschlangenlängen ist, dass alle lokalen Gutströme  $Z$  des Systems in ihrer Parameterdarstellung  $\boldsymbol{\pi}(Z) = (\pi_1(Z), \pi_2(Z), \pi_3(Z))$  vorliegen.

Es wird ferner angenommen, dass in jedem Fall bekannt ist, welchen Pfad eine Transporteinheit für die Erfüllung eines Transportauftrages nimmt. Aus den Transportmatrizen und den zugehörigen Pfadinformationen können dann an jeder Stelle des Systems die lokalen Transportmatrizen berechnet werden. Damit ist an jeder Stelle  $Z$  der mittlere Abstand zweier aufeinander folgender Transporteinheiten  $\pi_1(Z) = EZ$  berechenbar.

An jedem Ausgang eines Transportsystem-Elements ergibt sich der minimale Abstand  $\pi_2(Z) = t_{min}(Z)$  zwischen Transporteinheiten aus der konkreten technischen Ausführung des Elements. Diese Größen werden vom Hybridmodell nicht berechnet, sie sind Parameter, die der Nutzer zur Verfügung stellen muss.

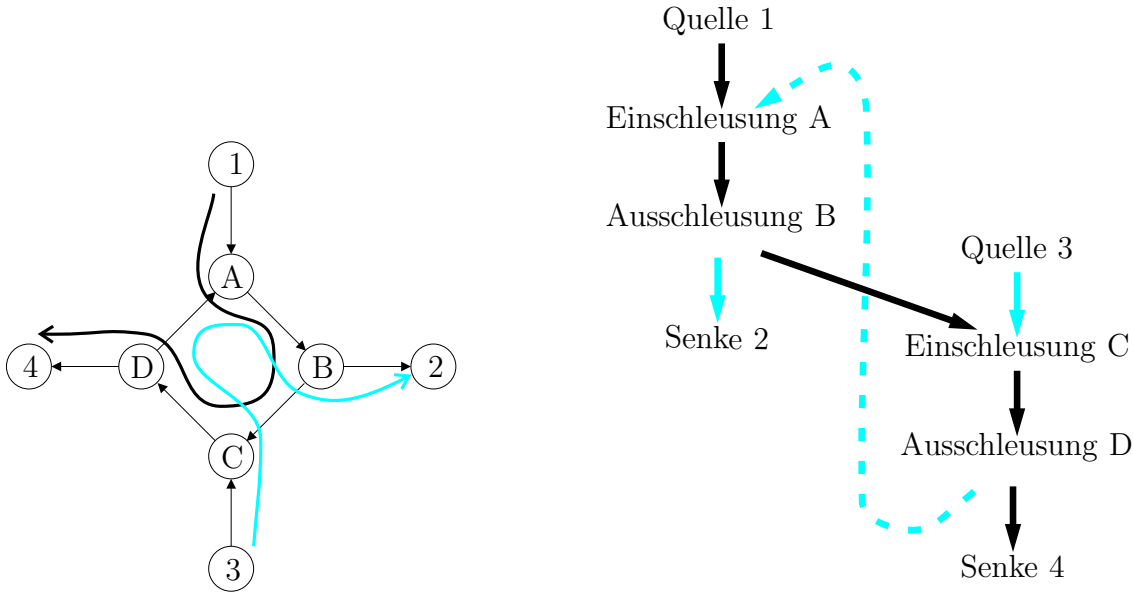
Auch für die Variationskoeffizienten der Eingangsströme des Systems muss der Nutzer eine Abschätzung treffen. Sind für ein Element oder Teilsystem die Parameter aller Eingangsströme bekannt, können alle seine Ausgangsparameter berechnet werden. Die Reihenfolge der Berechnung legt der Nutzer des Hybridmodells so fest, dass dies möglichst immer der Fall ist.

Das Modell berechnet daraufhin entsprechend der vorgegebenen Reihenfolge die statistischen Parameter der Elemente bzw. Teilsysteme. Für deren numerische Approximation werden die trainierten neuronalen Netze eingesetzt. Die Variationskoeffizienten der Abgangsströme  $\pi_3(Y_j)$  werden den nachfolgenden Bausteinen als Eingangsparameter zur Verfügung gestellt.

Als Ergebnis der Modellierung mit dem Hybridmodell stehen dann an jeder Stelle die berechneten Parameter der Gutströme und die interessierenden Parameter der Elemente bzw. Teilsysteme zur Verfügung.

### 5.1.2 Zyklische Berechnungsgraphen

Infolge der Überlagerung von Gutströmen ist es nicht immer möglich, einen zyklensfreien Berechnungsgraphen für die Variationskoeffizienten  $\pi_3(Z) = c_Z$  aufzustellen.



**Abbildung 5.3:** Beispiel für gegenseitige Abhängigkeit der Komponente, Aufbrechen des Berechnungsgraphen für zyklische Fälle

Abbildung 5.3 skizziert ein einfaches Beispiel dafür. Links werden die Gutströme skizziert: von der Quelle 1 werden die Transporteinheiten über die Elemente A, B, C und D zur Senke 4 transportiert. Gleichzeitig bewegen sich Transporteinheiten von Quelle 3 über die Elemente C, D, A und B zur Senke 2. Als Hauptstrecke wird der Zyklus A-B-C-D-A betrachtet.

Die skizzierten Ströme überlagern sich so, dass der Berechnungsgraph (Abbildung 5.3, rechts) nicht baumartig ist, sondern Zyklen besitzt. Das bedeutet, dass zum Beispiel der Variationskoeffizient des Ausgangsstromes des Elementes A vom Variationskoeffizienten des Ausgangsstromes des Elements C abhängt und umgekehrt. Um dieses Beispiel trotzdem mit dem hier vorgestellten Modell abbilden zu können, muss der Berechnungsgraph aufgebrochen werden. Dieses Vorgehen wird im Folgenden diskutiert.

Abbildung 5.4 zeigt den Abgangsstrom  $Y$  einer Komponente mit den Eingangsströmen  $X_i$ , also  $Y = f(X_1, X_2, \dots)$ . Das System habe eine solche Struktur, dass  $X_k$  abhängig von  $Y$  ist. Das heißt, der Rest des Transportsystems bestimmt eine Funktion<sup>1</sup>  $\xi(\cdot)$  mit  $X_k = \xi(Y)$ .

<sup>1</sup>Es soll an dieser Stelle nicht suggeriert werden, dass  $X_k$  einzig von  $Y$  abhängt. Diese Schreibweise wurde lediglich gewählt, um eine kompakte Darstellung zu ermöglichen. Wenn mehrere Eingänge der Komponente vom Ausgang  $Y$  abhängen, können diese Betrachtungen analog erweitert werden.



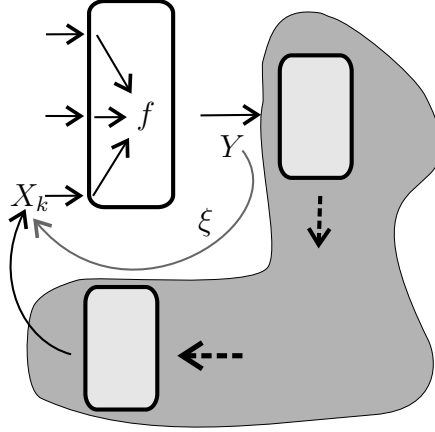


Abbildung 5.4: Abhängigkeiten in einem Zyklus

Dann gibt es einen Strom  $\tilde{Y}$  mit

$$\tilde{Y} = f(X_1, X_2, \dots, X_{k-1}, X_k, \dots) = f(X_1, X_2, \dots, X_{k-1}, \xi(\tilde{Y}), \dots), \quad (5.1)$$

das ist nämlich gerade der Zustand des stationären Systems.

Sei  $(Y_n) = (Y_0, Y_1, \dots)$  eine Folge von Ausgangsströmen mit

$$Y_{i+1} = g(Y_i) := f(X_1, X_2, \dots, X_{k-1}, \xi(Y_i), \dots). \quad (5.2)$$

**Lemma:** Die Folge  $(Y_n)$  aus Gleichung (5.2) konvergiert

$$\lim_{n \rightarrow \infty} Y_n = \hat{Y} \quad (5.3)$$

und es gilt  $\hat{Y} = \tilde{Y}$ .

Für den Beweis des Lemmas mit dem Fixpunktsatz von Banach (z.B. in [Heu92]), müsste gezeigt werden, dass in der Nähe des Fixpunktes  $\tilde{Y}$  die Ableitung  $\frac{\partial g(Y)}{\partial Y} \ll 1$  ist.

Die Berechnung der partiellen Ableitungen

$$\frac{\partial g}{\partial Y} = \frac{\partial f}{\partial X_k} \cdot \frac{\partial \xi}{\partial Y}$$

ist für das reale System im Normalfall nicht möglich. Da aber alle Komponenten des hybriden Modells algorithmisch vorliegen, wäre hier ein Beweis mit Hilfe des automatischen Differenzierens (siehe dazu etwa [Gri00]) für ein konkretes System möglich. Der allgemeine Nachweis der Gültigkeit des Lemmas wird hier nicht erbracht. Es kann jedoch heuristisch bestätigt werden: Der „Charakter“ der Abgangsströme wird im Wesentlichen von den Eigenschaften des betrachteten Elements oder Teilsystems bestimmt. Das gilt zumindest dann, wenn eine geringe Auslastung dies nicht verhindert. Die Variationskoeffizienten der Eingangsströme bestimmen hingegen stark die Längen der Warteschlangen an den Eingängen.

Als Beispiel wird noch einmal die Struktur in Abbildung 5.3, links betrachtet. Jeder der beiden Gutströme hat mittlere Zwischenzeiten von 7 s, die Transfers benötigen keine zusätzliche Zeit, die Förderabstände betragen  $t_{min} = 1$  s. An den Eintrittstellen in das System (Einschleusung A und C) seien die Ströme exponentialverteilt. Es sei nun die Einschleusung A betrachtet. Der Eingangsstrom auf der Hauptstrecke hängt über die anderen Transportsystem-Elemente vom Ausgang dieser Komponente ab. Sei  $(c_n)$  die Folge der Variationskoeffizienten, die für einen Startwert  $c_0$  durch wiederholte Iteration (Gleichung 5.2) für diesen Eingangsstrom berechnet wird.

Für die erste Berechnung wird der Variationskoeffizient der Hauptstrecke vor dem Eingang A auf einen willkürlichen Wert  $c_0$  gesetzt. Sukzessive werden daraufhin alle anderen Variationskoeffizienten berechnet. Insbesondere erhält man einen neuen Wert  $c_1$  für den Variationskoeffizienten der Hauptstrecke vor dem Eingang A. Tabelle 5.1 zeigt einige Berechnungsbeispiele<sup>2</sup>. Bereits nach dem zweiten Iterationsschritt wird der Fixpunkt erreicht. Die Simulation bestätigt ihn mit  $c = 1.0$ .

$c_0$	$c_1$	$c_2$
0.2	0.952	0.973
0.5	0.976	0.973
1.0	0.974	0.973
1.5	0.986	0.974

**Tabelle 5.1:** Auswirkungen von Startparametern auf die Berechnung

Damit konnte demonstriert werden, wie zyklische Berechnungsgraphen gehandhabt werden. Eine Berechnung der Parameter des hybriden Modells ist auf diese Weise somit in der Regel möglich.

Im Folgenden werden in drei Beispielen die hybriden Modelle mit diskreten Simulationsmodellen verglichen, um die Grenzen und Möglichkeiten dieses Modellierungsansatzes zu skizzieren.

<sup>2</sup>Die dritte Stelle nach dem Komma wird hier nur angegeben, um diese Berechnungsschritte zu verdeutlichen. Die Toleranz der neuronalen Netze liegt etwa bei 0.005, deshalb wäre es aus Sicht der Modellierung falsch, hier solche Genauigkeit zu suggerieren.

## 5.2 Beispiele

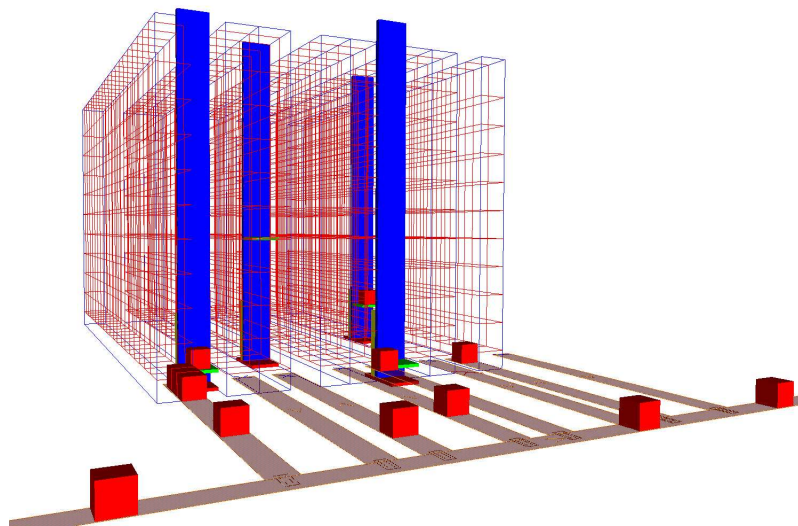
Es werden an dieser Stelle drei typische Systeme bzw. Teilsysteme betrachtet:

- Hochregallager mit vier Regalbediengeräten und Vorzone,
- Kommissionierbereich mit fünf Kommissionierplätzen,
- Kombination eines Hochregallagers mit einem Kommissionierbereich.

### 5.2.1 Hochregallager

#### Beschreibung der Transportsystem-Elemente

In diesem Beispiel wird die Anbindung eines Hochregallagers an ein Transportsystem untersucht. Hier kommen Dreh- bzw. Hubtische zum Einsatz, die einen rechtwinkligen Anschluss der Lagervorzone an eine Hauptstrecke ermöglichen (Abbildung 5.5).



**Abbildung 5.5:** Lagervorzone in AutoMod-Simulation

Modellsignifikante Parameter der Fördertechnik sind die folgenden:

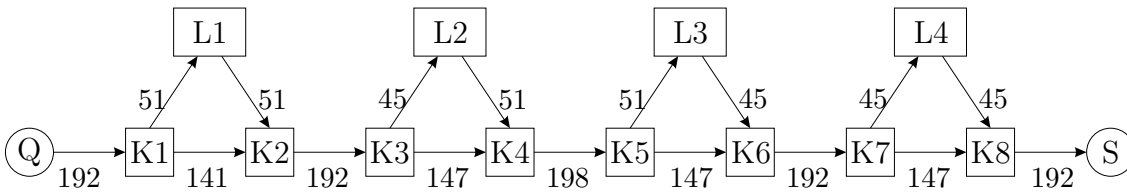
- Die Hubtische haben die folgenden Parameter: Transferzeit  $t_{tr} = 1$  s, Zeit für Heben bzw. Senken:  $t_m = 1$  s. Für die Steuerung wurde festgelegt, dass die Hauptstrecke Vorfahrt hat.
- Die Regale haben die gleiche Höhe  $H = 15$  m (10 Fächer je 1.5 m) und Länge  $L = 30$  m (20 Fächer je 1.5 m).
- Die Regalbediengeräte haben die gleichen Geschwindigkeiten  $v_v = 1 \text{ ms}^{-1}$ ,  $v_h = 2 \text{ ms}^{-1}$ .
- Für beide Bewegungsrichtungen ist die Beschleunigung  $a = 1 \text{ ms}^{-2}$ .
- Die Transferzeit für die Aufnahme bzw. Abgabe von Transporteinheiten beträgt  $t_{tr} = 5$  s.
- Die Positionierzeit beträgt  $t_p = 2$  s.
- Für das hybride Modell werden stattdessen entsprechend die Parameter  $T_h = 15$  s,  $T_v = 15$  s,  $t_p = 3.5$  s,  $t_{tr} = 5$  s benutzt.

## Gutströme

Der minimale Abstand zwischen den Transporteinheiten ist  $t_{min} = 1$  s. Mit den Bezeichnungen Q, S, L für Quelle (Eingang des Teilsystems), Senke (Ausgang des Teilsystems), bzw. Lager sei die folgende Transportmatrix zugrunde gelegt:

	Q	L1	L2	L3	L4	S
Q	0	51	45	51	45	0
L1	0	0	0	0	0	51
L2	0	0	0	0	0	51
L4	0	0	0	0	0	45
L5	0	0	0	0	0	45
S	0	0	0	0	0	0

Die Auslagerungen folgen einer Exponentialverteilung, der Eingangsstrom des Systems (an der Quelle Q) habe einen Variationskoeffizienten von  $c = 0.8$ . Die entsprechenden Streckenlasten werden in Abbildung 5.6 dargestellt.

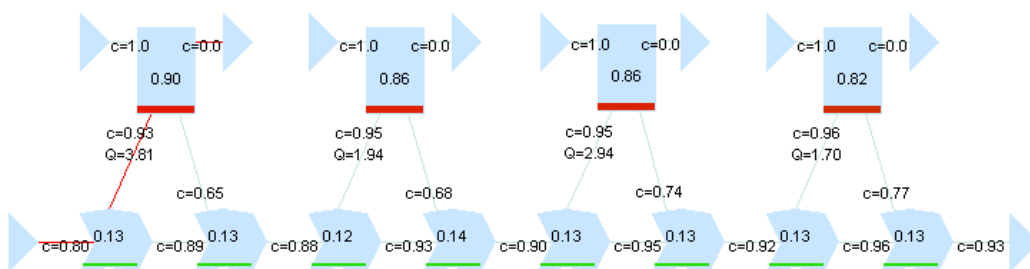


**Abbildung 5.6:** Lager mit Vorzone. Streckenbelastungen in Transporteinheiten pro Stunde. Die Hubtische werden mit  $K_1, \dots, K_8$  bezeichnet.

Aus den Streckenlasten ergeben sich die für die Modellierung benötigten lokalen Transportmatrizen für die Kreuzungen  $K_1, \dots, K_8$ .

## Vergleich mit diskreter Simulation

Diskrete Simulation und Hybridmodell zeigen einige Unterschiede, die im Folgenden diskutiert werden. Es werden in diesem Kapitel wiederholt Visualisierungen der Software „TransHyb“ für die hybride Modellierung gezeigt (vgl. Anhang A.3, S. 93f.). Diese prototypische Implementierung gestattet es, Strukturen nach den hier beschriebenen Prinzipien aufzubauen und zu berechnen. Bei den Darstellungen handelt es sich um die strukturierte Ausgabe von Ergebnissen. Die dabei verwendeten Symbole gehen meist aus dem Kontext hervor, eine Legende findet sich jedoch auch im Anhang auf Seite 94.



**Abbildung 5.7:** Hybrides Modell der Lagervorzone

Abbildung 5.7 zeigt das Ergebnis der Berechnung des Hybridmodells. Es werden jeweils die berechneten 90%-Quantile der Warteschlangen (wenn sie größer als Null sind) und die Variationskoeffizienten der entsprechenden Strecken, sowie die Auslastungen der Transportsystem-Elemente dargestellt.

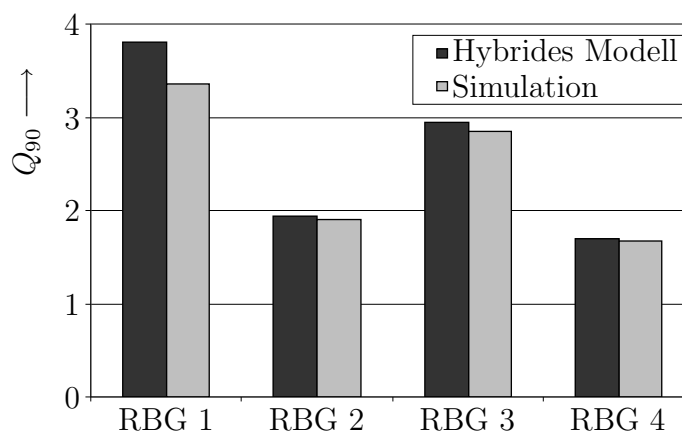
Die Regalbediengeräte werden mit jeweils zwei Ein- und Ausgängen abgebildet (vgl. S. 50). Die Ströme der Auslagerungsaufträge werden im Hybridmodell mit denselben Parametern abgebildet wie Gutströme. Die Ströme der Einlagerungsbestätigungen sind im Rahmen dieser Arbeit wenig interessant. Ihre Variationskoeffizienten werden deshalb nicht berechnet. An entsprechender Stelle ist in Abbildung 5.7 darum „c=0“ zu lesen.

Variationskoeffizienten: Hier interessiert zunächst der Parameter des Ausgangsströms des gesamten Systems. Die diskrete Simulation liefert 0.88, das hybride Modell 0.93.

Die Eingänge zu den Regalbediengeräten stimmen in beiden Modellen bis auf 0.01 überein. Bei den Ausgängen sind die Unterschiede ebenso klein.

Auslastungen: Die Auslastungen der Regalbediengeräte und der Hub- bzw. Drehtische weichen bei beiden Modellen weniger als 0.015 ab.

90%-Quantile der Warteschlangenlängen: Wichtigstes Ziel der Modellierung ist jedoch, den Staubereich der Lagervorzone (Einlagerung) zu prognostizieren. Hier liefert die Simulation die folgenden Werte (vgl. Abbildung 5.7, von links): 3.36, 1.90, 2.85, 1.67. In Abbildung 5.8 werden die Warteschlangenquantile aus beiden Modellierungen gegenübergestellt.



**Abbildung 5.8:** Vergleich der Warteschlangenquantile vor den Regalbediengeräten

Offenbar weicht das Hybridmodell für die Prognose längerer Warteschlangenquantile stärker von der Simulation ab. Ein Grund dafür ist, dass für diese Bereiche vergleichsweise wenig Trainingsdaten erzeugt wurden. Mit Sicherheit kann aber das neuronale Netz für die Vorhersage des Quantils mit Hilfe weiterer Trainingsdaten an eine erhöhte Auslastung angepasst werden.

### 5.2.2 Kommissionierbereich

Ein Kommissionierbereich mit fünf Arbeitsplätzen wird an das Transportsystem (Hauptstrecke) in der folgenden Weise angebunden (Abbildung 5.9):

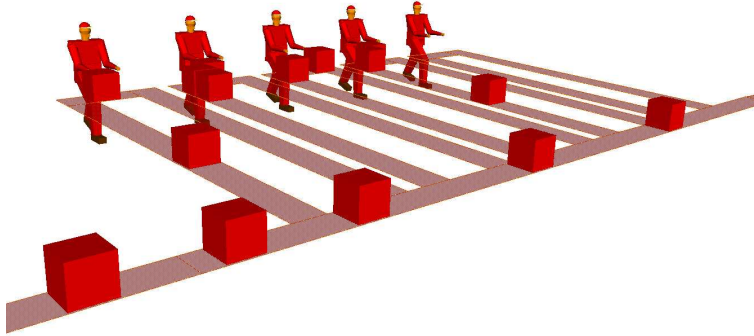


Abbildung 5.9: Kommissionierbereich im AutoMod-Modell

- Die maximale Kommissionierleistung betrage an allen Plätzen 180 Transporteinheiten pro Stunde.
- An den Kommissionierplätzen wird unterschiedlich gleichmäßig gearbeitet. Die Variationskoeffizienten der dabei zugrunde liegenden Gammafunktion für die Kommissionierplätze Ko1,..., Ko5 seien

Ko1	Ko2	Ko3	Ko4	Ko5
0.4	0.6	0.8	1.0	1.2

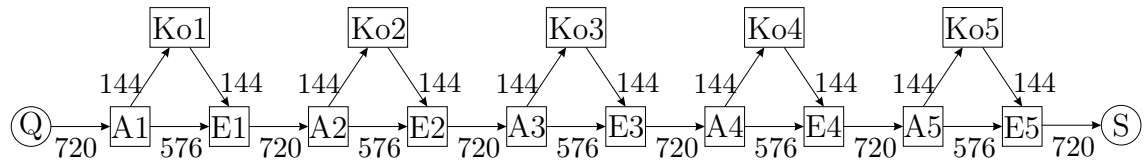
- Ausschleusungen und Einschleusungen haben Transferzeiten von 1 s.

#### Gutströme

Alle 5 s trifft ein Kommissionierauftrag für einen der Kommissionierer am Systemeingang ein (gleichverteilte Last). Die Transporteinheiten bewegen sich zum Kommissionierer und nach der Bearbeitung zum Ausgang des (Teil-)Systems. Es ergibt sich dann die folgende Transportmatrix:

	Q	Ko1	Ko2	Ko3	Ko4	Ko5	S
Q	0	144	144	144	144	144	0
Ko1	0	0	0	0	0	0	144
Ko2	0	0	0	0	0	0	144
Ko3	0	0	0	0	0	0	144
Ko4	0	0	0	0	0	0	144
Ko5	0	0	0	0	0	0	144
S	0	0	0	0	0	0	0

Der Variationskoeffizient des Gutstroms am Systemeingang betrage  $c = 0.7$ . Abbildung 5.10 zeigt die resultierende Streckenlast.

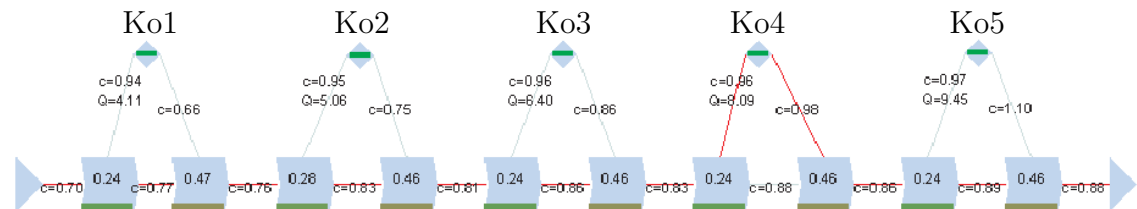


**Abbildung 5.10:** Kommissionierbereich. Streckenbelastungen in Transporteinheiten pro Stunde. Die Ein- und Ausschleusungen von der Hauptstrecke werden mit E bzw. A bezeichnet.

Aus den Streckenlasten ergeben sich die für die Modellierung benötigten lokalen Transportmatrizen für die Ein- und Ausschleusungen  $E_1, \dots, E_4$  bzw.  $A_1, \dots, A_4$ .

### Vergleich mit der diskreten Simulation

Eine diskrete Simulation zur Überprüfung der Daten des hybriden Modells (5.11) ergab eine sehr hohe Übereinstimmung.

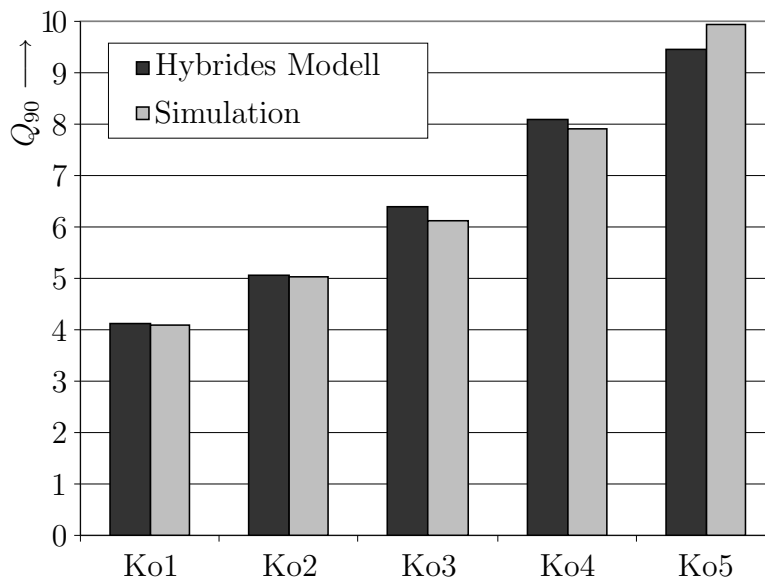


**Abbildung 5.11:** Kommissionierbereich

Variationskoeffizienten: Es reicht fast aus, den Ausgangsvariationskoeffizienten des Systems in beiden Modellen nebeneinander zu stellen: das Hybridmodell errechnet 0.88, die Simulation 0.89! Die Ausgänge der einzelnen Ausschleusungen sind ähnlich genau.

Bei den Ausgängen der Kommissionierer gibt es ebenfalls nur marginale Abweichungen. Die größte Abweichung ist bei dem letzten Kommissionierer (Ko5) zu finden. Hier sind die Variationskoeffizienten der Abgangsströme 1.10 (Hybridmodell) bzw. 1.12 (Simulation).

90%-Quantile der Warteschlangenlängen: Da der Querverkehr auf der Hauptstrecke verhältnismäßig gering ist, kommt es an den Einschleusungen nicht zu Staus. Darin stimmen diskrete Simulation und das Hybridmodell überein. Vor den Kommissionierern ermittelt die diskrete Simulation die folgenden Warteschlangenquantile: 4.09, 5.04, 6.11, 7.91, 9.94. In Abbildung 5.12 werden die Warteschlangenquantile aus beiden Modellierungen gegenübergestellt.



**Abbildung 5.12:** Vergleich der Warteschlangenquantile vor den Kommissionierplätzen

Dieser Kommissionierbereich wird von der Simulation sehr gut abgebildet. Das liegt aber nicht an der geschickten (oder gar gezielten) Auswahl der Parameter, sondern schlicht daran, dass die auftretenden Verteilungen an den Eingängen der einzelnen Transportsystem-Elemente dicht an der Gammaverteilung liegen:

- Die Kommissionierzeit lässt sich in der Realität nur sehr schwer abschätzen. Hier eine Gammaverteilung anzusetzen scheint legitim.
- Die Zwischenzeiten der Aufträge für die einzelnen Kommissionierer sind annähernd exponentialverteilt.
- Die Ausgänge der Kommissioniererplätze sind somit jeweils als Summe von Gammaverteilungen wieder gammaverteilt.

Die Ausgangsströme der einzelnen Transportsystem-Elemente des Systems gehorchen demnach etwa Gammaverteilungen. Sie entsprechen damit direkt den Trainingsdaten der neuronalen Netze. Dadurch bewegen sich die Prognosen in einem sehr sicheren Kernbereich.

Für einen Kommissionierbereich mit solch einem Design ist das hybride Modell sehr gut geeignet<sup>3</sup>.

<sup>3</sup>Weitere, hier nicht dokumentierte Experimente haben gezeigt, dass diese Aussage auch zutrifft, wenn zusätzlich ein hoher Verkehr auf der Hauptstrecke angenommen wird.



### 5.2.3 Gesamtsystem mit Lager und Kommissionierbereich

#### Beschreibung der Transportsystem-Elemente

Abschließend wird eine typische Konfiguration aus Hochregallager und Kommissionierzone mit Hilfe des hybriden Modells abgebildet. Das Modell ist an eine aktuelle Simulationsstudie angelehnt (Abbildung 5.13).

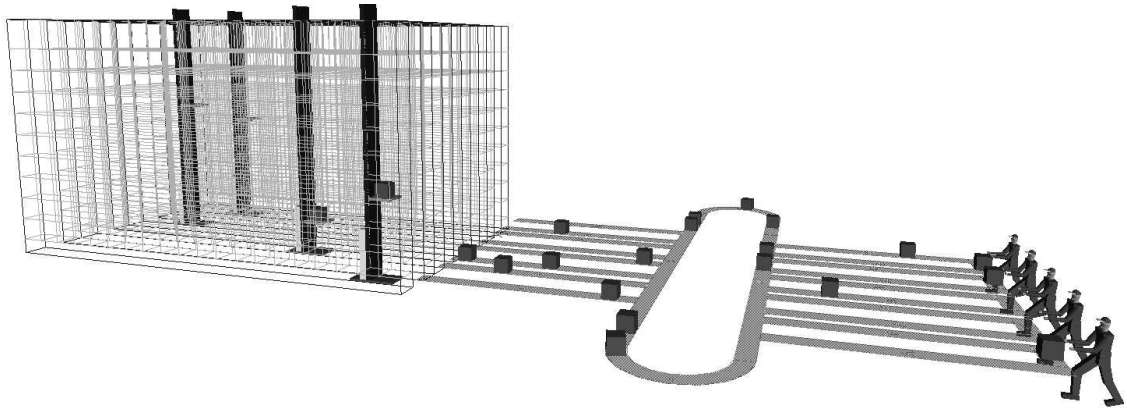


Abbildung 5.13: Gesamtsystem als AutoMod-Modell

Für die Kombination der beiden Teilsysteme aus den vorangegangenen Abschnitten sollen einige Parameter angepasst werden, um hier zu interessanten Beobachtungen zu gelangen. Die fördertechnischen Parameter der Regalbediengeräte bleiben gleich, aber die Kommissionierzeiten werden verändert, um größere Warteschlangen vor den Kommissionierplätzen zu beobachten. Die Bedienzeiten  $T_B$  der Kommissionierer Ko1,..., Ko5 sind gammaverteilt mit den folgenden Parametern (vgl. Abbildung 5.14):

	Ko1	Ko2	Ko3	Ko4	Ko5
$EB$	70 s	60 s	55 s	50 s	50 s
$c_B$	0.7	1.0	0.4	1.0	1.3

#### Gutströme

Im System gibt es die folgenden Gutströme:

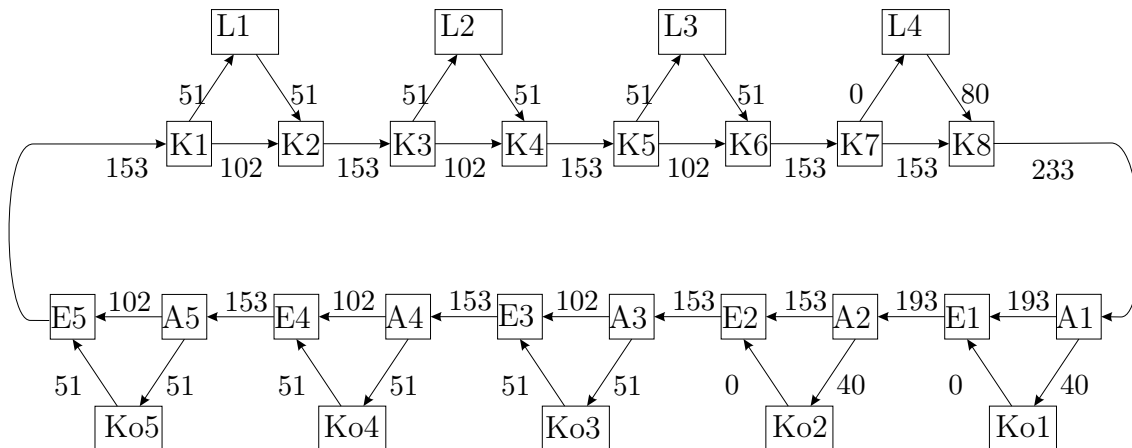
- von L1 zu Ko3 und wieder zurück:  $c = 1.0$ ,  $EX = 70$  s,
- von L2 zu Ko4 und wieder zurück:  $c = 1.0$ ,  $EX = 70$  s,
- von L3 zu Ko5 und wieder zurück:  $c = 1.0$ ,  $EX = 70$  s,
- von L4 zu Ko1:  $c = 1.0$ ,  $EX = 90$  s sowie
- von L4 zu Ko2:  $c = 1.0$ ,  $EX = 90$  s.

Das entspricht der folgenden Transportmatrix:

	L1	L2	L3	L4	Ko1	Ko2	Ko3	Ko4	Ko5
L1	0	0	0	0	0	0	51	0	0
L2	0	0	0	0	0	0	0	51	0
L3	0	0	0	0	0	0	0	0	51
L4	0	0	0	0	40	40	0	0	0
Ko1	0	0	0	0	0	0	0	0	0
Ko2	0	0	0	0	0	0	0	0	0
Ko3	51	0	0	0	0	0	0	0	0
Ko4	0	51	0	0	0	0	0	0	0
Ko5	0	0	51	0	0	0	0	0	0

Die daraus resultierende Belastung der einzelnen Strecken wird in Abbildung 5.14 dargestellt.

**Bemerkung:** Obwohl einige Strecken in dieser Studie nicht benutzt werden, sind sie im Materialfluss-System vorhanden. Das betrifft die Lagervorzone (Einlagerung) des Lagers L4 und die Verbindungen der Kommissionieplätze Ko1 und Ko2 zurück zur Hauptstrecke.

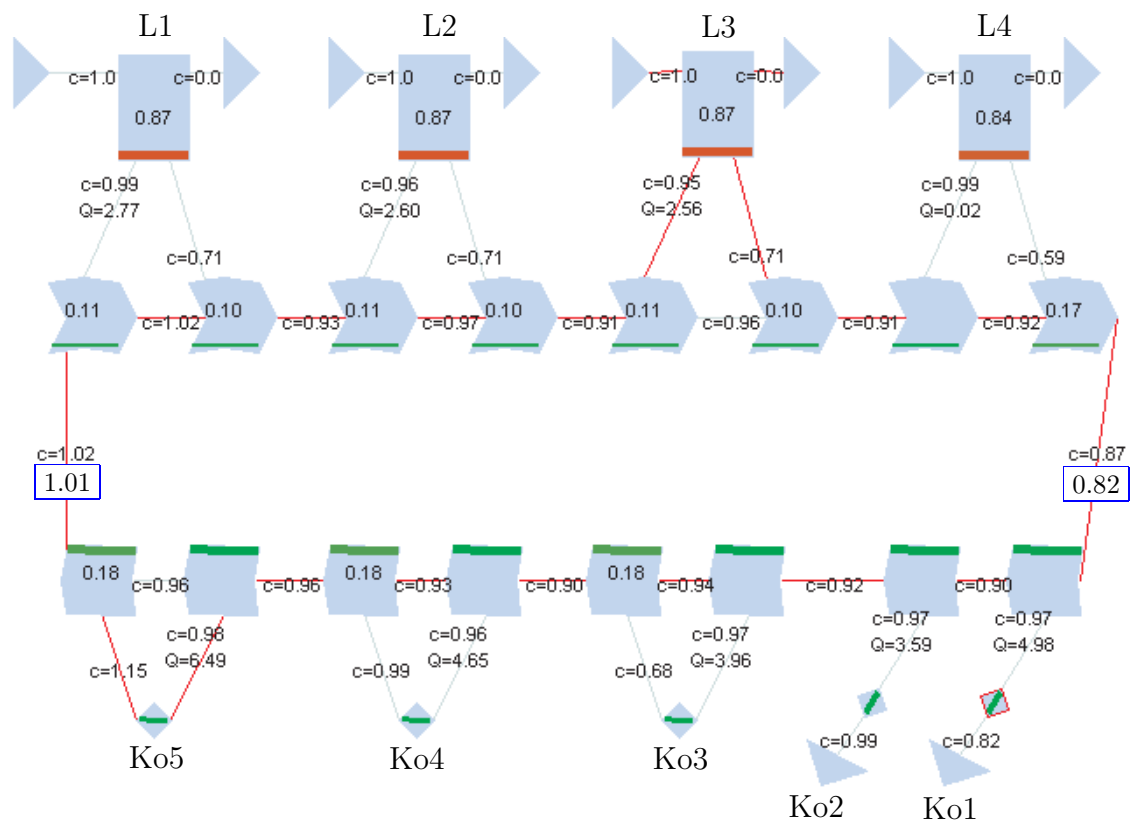


**Abbildung 5.14:** Gesamtsystem. Streckenbelastungen in Transporteinheiten pro Stunde.

## Vergleich mit diskreter Simulation

Variationskoeffizienten: In Abbildung 5.15 werden die Ergebnisse des Hybridmodells dargestellt. Eingerahmt findet man zwei Variationskoeffizienten der diskreten Simulation. Die Abweichung auf dem Weg zwischen dem Lager und den Kommissionierern beträgt 0.05, das ist auch die maximale Abweichung im System.

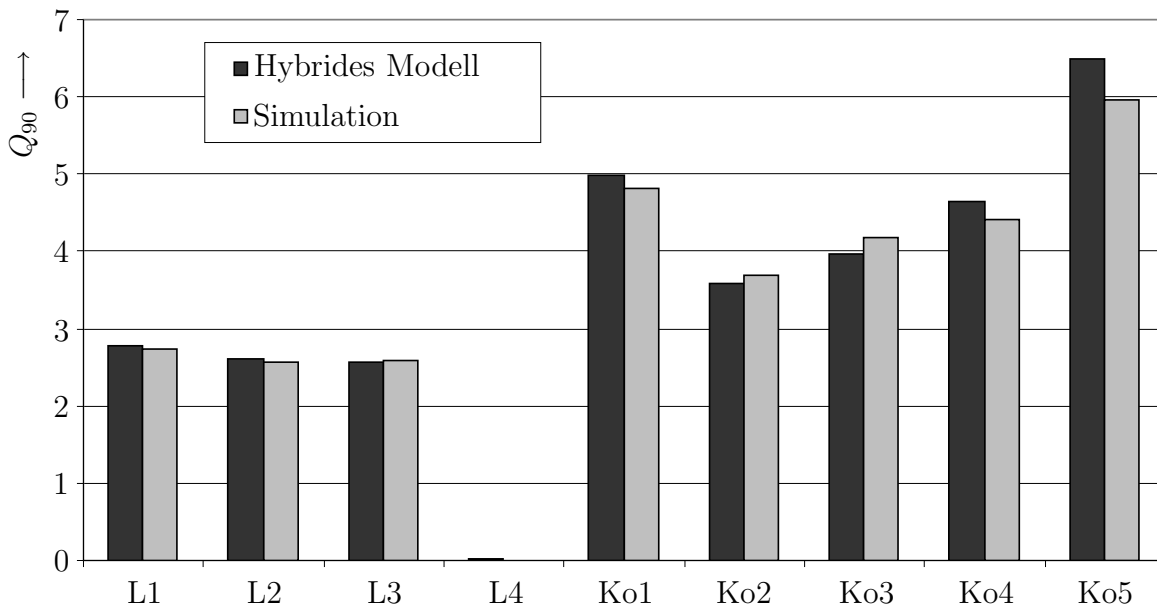
Zur Berechnung des hybriden Modells wurde der Berechnungsgraph (links) an der Stelle zwischen dem Kommissionierbereich und dem Hochregallager aufgetrennt. *De facto* unabhängig von der Festlegung des Induktionsanfanges (vgl. Diskussion auf S.75f.) wurde nach 3 Iterationsschritten ein Fixpunkt erreicht.



**Abbildung 5.15:** Gesamtsystem. Ergebnisse der hybriden Modellierung

Auslastung: Die Abweichungen der Auslastungen liegen für alle Elemente unter 0.02.

Quantile der Warteschlangenlängen: Abbildung 5.16 stellt die Ergebnisse beider Berechnungsmethoden gegenüber. Im Bereich großer Warteschlangenlängen sind wieder etwas größere Abweichungen zu beobachten, aber sicher kann dieser Effekt durch gezieltes Training der neuronalen Netze verringert, wenn nicht gar beseitigt werden.



**Abbildung 5.16:** Vergleich der Warteschlangenquantile

Damit wird deutlich, dass das hybride Modell eine sehr gute Näherung für dieses Transportsystem berechnet.

### 5.3 Zusammenfassung

Systematische Untersuchungen zur Güte der Prognosen sind schwer möglich, da sowohl die Struktur des Transportsystems, als auch die Parameter der einzelnen Elemente bzw. Teilsysteme verändert werden können. Und selbst wenn es eine Methode dafür gäbe, kann trotzdem eine Bedienzeitverteilung einer Bearbeitungsmaschine gefunden werden, für die dieses Modell nicht funktioniert. Das Ziel der aufgezeigten Beispiele konnte daher nur darin liegen, die Fähigkeiten der hybriden Modellierung zu demonstrieren. Wenn erst einmal ein gewisses Vertrauen in diese Methode entstanden ist, können weitere Experimente zur Erprobung des Modells in der Praxis erfolgen.

# 6 Fazit

## 6.1 Ergebnisse der Arbeit

In den vorangegangenen Kapiteln wurde eine neue Möglichkeit für die Modellierung von Materialfluss-Systemen vorgestellt. Mit Hilfe des hybriden Modells können die folgenden statistischen Kenngrößen für jedes hier untersuchte fördertechnische Element prognostiziert werden:

- Auslastung  $\varrho$ ,
- 90%-Quantil der Warteschlangen vor den Eingängen,
- Variationskoeffizienten der Abgangsströme.

Für die untersuchten Modellklassen wird in Tabelle 6.1 eine Übersicht über die Genauigkeit der Modelle gegeben. Dabei steht  $\varepsilon_{90}(X)$  für das 90%-Quantil des absoluten Fehlers bei der Modellierung des Parameters<sup>1</sup>  $X$ :

Element	Ein-/Ausgänge	$\varepsilon_{90}(c)$	$\varepsilon_{90}(Q_{90})$	$\varepsilon_{90}(\varrho)$
Einschleusung	2/1	0.05	0.2	-
Ausschleusung	1/2	0.02	0.2	-
Kreuzung	2/2	0.05	0.3	0.015
Regalbediengerät	2/2	0.03	0.2	0.010
Bedienstation	1/1	0.02	0.3	-

**Tabelle 6.1:** Modellierte Elemente und die 90%-Quantile der Fehler (Ergebnisse der Verifikation)

Trotz einiger Einschränkungen ermöglicht das Hybridmodell eine Planungssicherheit, die für einen Grobentwurf des Transportsystems weit über die gebräuchlicher statischer Berechnungsmodelle hinausgeht (vgl. Kapitel 2.3).

## 6.2 Ausblick

### 6.2.1 Modellierung weiterer Klassen

Die beschriebene Methode kann angewandt werden, um weitere Steuerungsregeln oder weitere Transportsystem-Elemente zu modellieren:

- Kreuzungen mit anderen Parkstrategien oder mit anderen Vorfahrtsregeln,
- Regalbediengeräte mit anderen Parkstrategien,

---

<sup>1</sup>Gibt es im betrachteten Element mehrere Parameter  $X$  wird der maximale Fehler angegeben.

- Regalbediengeräte mit Ein- und Auslagerungspunkten in verschiedenen Höhen,
- Karussellager,
- Regalbediengeräte mit Mehrfachlastaufnahme-Mitteln und verschiedenen Batch-Strategien,
- Verteilwagen u.a.

Da die Modellierung einer neuen Klasse mit einem gewissen Aufwand verbunden ist, empfiehlt es sich, diese neuen Klassenmodelle erst bei Bedarf zu erstellen.

### 6.2.2 Modellparameter

Sowohl die hier dokumentierten Beispiele als auch weitere Experimente mit dem hybriden Modell zeigen deutlich, dass es in der Lage ist, sehr gute Prognosen zu liefern. Die Aussagen des Modells können unter Umständen verbessert werden, wenn zur Charakterisierung der Ströme weitere Parameter hinzugenommen werden, zum Beispiel weitere zentrale Momente der Verteilungsfunktion.

Eine anderer Ansatz wäre, statt einzelner Parameter die Verteilungsfunktionen der Gutströme selber zu approximieren. Dabei ist es wichtig, die Anzahl der beschreibenden Parameter klein zu halten, denn für jeden neuen Parameter wird ein weiteres neuronales Netz benötigt. Aufgrund des Verlaufs der Funktion könnten auch dafür neuronale Netze benutzt werden. Der Parametervektor  $\pi(Z)$  eines Stromes  $Z$  würde dann für eine feste Netztopologie die Kantengewichte und Parameter der Aktivierungsfunktion beinhalten. - Neuronale Netze würden dann neuronale Netze berechnen.

Auf der anderen Seite darf, wie oben erwähnt, die „Verfälschung“ der Ergebnisse durch Blockierungen und Steuerungen nicht unterschätzt werden. Auch wenn das Modell durch den Einsatz weiterer Parameter für die Charakterisierung der Ströme korrekter im Sinne der Modellannahmen wird, heißt das nicht, dass es reale Größen besser prognostiziert. Ein Handlungsbedarf für die Erweiterung des Parameterraumes besteht erst dann, wenn die Ergebnisse der Modellierung zu stark von realen (oder simulierten) Daten abweichen, obwohl noch nicht allzu weit von den Modellannahmen abgerückt wurde.

Weitere interessante statistische Kenngrößen der Materialfluss-Elemente sind:

- mittlere Warteschlangenlänge, weitere Quantile anderer Ordnung,
- Wartezeit und Bedienzeit (mittlere und maximale Zeiten, Quantile),
- Energieverbrauch, Anzahl der Schaltvorgänge und der Bewegungen mit und ohne Last.

Trainingsdaten für weitere Quantile ( $Q_{75}$ ,  $Q_{95}$ ) und mittlere Warteschlangenlängen liegen bereits in aufbereiteter Form vor. Für die anderen Größen können die Daten entsprechend dem beschriebenen Vorgehen durch Simulationen erzeugt werden. Für jede Kenngröße und jede Modellklasse müssen dann weitere Netze mit diesen Daten trainiert werden.

Wie bereits mehrfach in der Arbeit erwähnt wurde, werden die Stützstellen zum Training der neuronalen Netze zufällig gleichverteilt im Parameterraum gewählt. Spätere Untersuchungen können gegebenenfalls von dieser Gleichverteilung der Parameter-Stützstellen im Parameterraum abgehen, um beispielsweise gezielt sehr hohe Warteraumbelegungen zu betrachten. Dies sollte jedoch erst dann geschehen, wenn die gebauten hybriden Modelle aus diesem Grund in der Praxis versagen. Eine Voraussetzung für die Berücksichtigung einer Ungleichverteilung ist die Kenntnis einer praxisrelevanten Verteilung der Parameter im Parameterraum.

### 6.2.3 Weitere Ideen

Das in Kapitel 5 beschriebene hybride Modell ist in seinem Einsatz nicht auf Materialfluss-Systeme beschränkt. Mit Hilfe geeigneter Metamodelle für Einzelkomponenten sind hiermit ggf. auch Prognosen für *Supply chains* bzw. *Supply networks* möglich.

Weitere interessante Erkenntnisse über die Robustheit eines hybriden Modells gegenüber der Änderung einzelner Parameter können mit Hilfe des automatischen Differenzierens gewonnen werden. Dazu gehört beispielsweise die Aussage, bei welchen Parameterkonfigurationen die Länge einer Warteschlange zu stark ansteigt für eine sichere Prognose. Für einen beliebigen Systemparameter  $\alpha$  (egal ob logistischer oder technischer Art) können damit auch für Warteschlangenquantile  $Q_{90}(Z)$  die partiellen Ableitungen  $\partial Q_{90}(Z)/\partial \alpha$  berechnet werden. Damit können kritische Systemparameter aufgespürt werden, bei denen geringe Änderungen reichen, um das gesamte System empfindlich zu stören.

Bislang wird für die Berechnung des hybriden Modells vorausgesetzt, dass der Berechnungsgraph (manuell) gegeben wird. Hier können, auch in Erweiterung der Stabilitätsuntersuchungen, geeignete Methoden aus der Graphentheorie benutzt werden, um den Graphen automatisch zu generieren. Dazu gehört insbesondere das Erkennen und Aufbrechen von Zyklen in den Berechnungsgraphen.

# Anhang A

## Benutzte Werkzeuge

### A.1 Erzeugung der Trainingsdaten

Um eine statistische Kenngröße einer Klasse von Materialfluss-Elementen mit einem neuronalen Netz zu approximieren, muss der interessierende Wert (etwa die Auslastung des Elements) zunächst für eine große Menge von Vertretern dieser Klasse ermittelt werden. Dabei müssen die numerischen Parameter des Elements und der lokalen Gutströme über dem gesamten (sinnvollen) Parameterraum variiert werden. Wir wählen sie zufällig gleichverteilt aus dem Parameterraum aus.

Mit Hilfe der diskreten Simulation wird für jedes Parametertupel ein Simulationslauf durchgeführt. Die Dauer des Laufes wird während der Simulation dynamisch bestimmt: wenn die Änderung der interessanten Werte (Quantile der Warteschlangen-, Variationkoeffizienten der Abgangsströme) eine gewisse Schwelle unterschreitet, wird die Simulation abgebrochen. Simulationsläufe für Regalbediengeräte laufen über etwa 50 Tage. Da aber die Bedienzeiten dabei sehr groß sind, verbraucht solch ein Lauf etwa die gleiche Rechenzeit, wie eine Simulation einer Bedienstation über 10 Tage (s.a. Bemerkung S. 36).

Da diese Simulationen etwa 4000 Mal durchgeführt werden (3000 zum Training des Netzes, 1000 für die Verifikation), ist es essentiell, die benötigte Rechenzeit zu minimieren. Obwohl kommerzielle (AutoMod von Brooks PRI) und freie Simulatoren ([Bie95], [Lec99], [Bor94]) zur Verfügung stehen, wurde ein eigener Simulator benutzt. Diese Java-Software berechnet die benötigten Daten etwa 10 Mal schneller als AutoMod. Das liegt insbesondere daran, dass die gesamte Rechenzeit bei dem eigenen Simulator einzig zum Generieren der Trainingsdaten benutzt wird und keine zusätzlichen Leistungen (nicht benötigte erweiterte Statistiken, Protokolle etc.) bereitgestellt werden. Für die Erzeugung der Zufallsgrößen und die statistischen Auswertungen wurde die Java-Bibliothek *PSOL* (siehe nächster Abschnitt) eingebunden. Die Berechnung<sup>1</sup> für einen Parametertupel benötigt damit zwischen 5 und 20 s Rechenzeit. Die für den Betrieb des Simulators nötige Software (Java 2 Runtime Environment ab Version 1.4) ist bei Sun Microsystems (<http://java.sun.com>) frei verfügbar. Herkömmliche Internet-*Browser*, wie etwa Mozilla, enthalten sie bereits im Standardumfang.

---

<sup>1</sup>Sie Berechnungen erfolgten auf Sparc20, PCs mit Athlon 1600+ bzw. Dual Pentium 2 x 1 GHz.



### A.1.1 Statistik-Bibliothek

Für die direkte Auswertung der statistischen Daten konnte die Java-Bibliothek *Probabilistic/Statistic Object Library* der University of Alabama, Huntsville (USA) für den diskreten Simulator benutzt werden [Sie02]. Hier werden Objekte für die Erzeugung von Zufallsgrößen, für umfangreiche Datenerfassung und für die Auswertung der Daten bereitgestellt.

Der Autor konnte einen bescheidenen Beitrag leisten, um dieses *Open-Source*-Projekt weiterzuentwickeln. Dazu gehören ein modifiziertes Gradientenverfahren zur Ermittlung von gammaverteilten Zufallszahlen sowie die Implementierung der verstetigten Quantile.

## A.2 Entwurf und Training der neuronalen Netze

Mit dem *Stuttgart Neural Network Simulator* SNNS steht ein mächtvolles Werkzeug für die Arbeit mit neuronalen Netzen zur Verfügung [Zel95]. Es gestattet das Design und das Training neuronaler Netze mit diversen Trainingsmethoden.

Der Simulator läuft auf Unix-Workstations und auf PCs. Da er aber eine X-Windows-Benutzerschnittstelle verwendet, ist für MS-Windows-Rechner ein X-Windows-Emulator notwendig. Als Alternative dazu steht die Java-Version JNNS mit einer etwas eingeschränkten Funktionalität zur Verfügung (Abbildung A.1).

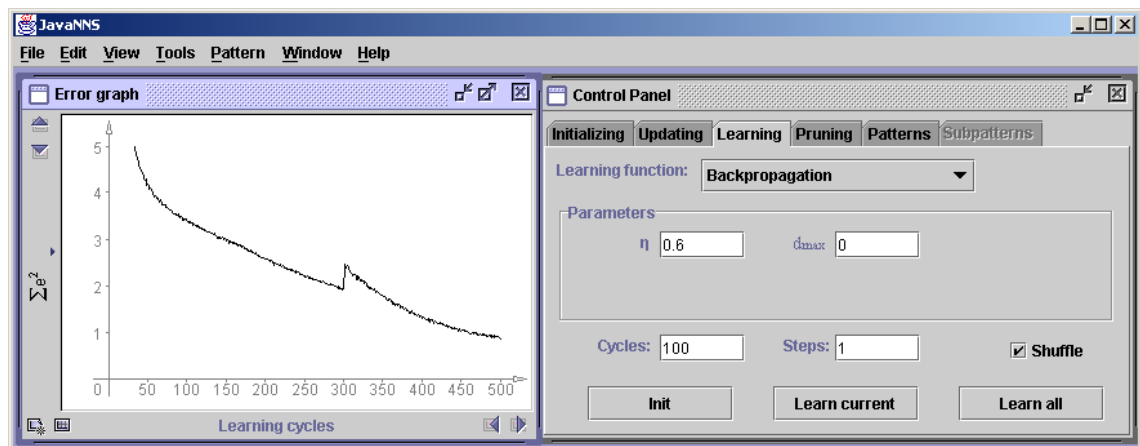


Abbildung A.1: JNNS - Bedienoberfläche

### A.2.1 Topologie der Netze

Die Anzahl der Eingangsneuronen entspricht der Anzahl der Eingangsgrößen des Approximationsproblems. In den hier betrachteten Fällen wurden nur Netze mit einem Ausgangsneuron benutzt. – Dadurch können die Zielgrößen (Warteschlangenquantile, Variationskoeffizienten und Auslastungen) jeweils unabhängig voneinander mit gewünschter Genauigkeit approximiert werden.

Das Programm SNNS gestattet das Design neuronaler Netze auf graphische und auf dialogbasierte Art und Weise. Die Auswahl der Aktivierungsfunktionen ist dem

Nutzer überlassen. Es empfiehlt sich jedoch, für die vorliegenden numerischen Probleme auf Funktionen zu verzichten, die nicht stetig differenzierbar sind.

### A.2.2 Training der Netze

Für das Training eines Netzes werden je eine Datei mit Trainings- und mit Verifikationsdaten ausgewählt. Die Größe des Dämpfungsp Parameters  $0 < \eta < 1$  bestimmt die Schrittweite zur Anpassung der internen Netzwerk-Parameter und damit indirekt die Geschwindigkeit bzw. Konvergenz des Trainings. Für das Training der in dieser Arbeit vorgestellten Netze wurde der Parameter beginnend mit  $\eta = 0.5$  schrittweise bis auf  $\eta = 0.01$  verkleinert.

Ein Trainingszyklus umfasst die Bewertung aller Trainingsdaten. Solche Zyklen müssen oft wiederholt werden. In einem graphischen Fenster kann die Größe des Netzfehlers in Abhängigkeit von der Zahl der absolvierten Trainingszyklen angezeigt werden. Es ist ebenfalls möglich, sich dort auch den Fehler der Approximation der Validierungsdaten anzeigen zu lassen: Ein gutes Indiz für ein *overfitting* ist es, wenn der Fehler für die Trainingsdaten kleiner wird, während die Validierungsdaten immer schlechter approximiert werden.

Die vorliegenden Ergebnisse dieser Arbeit wurden mit Hilfe des einfachen *backpropagation*-Verfahrens (vgl. Abschnitt 3.2) erzielt. SNNS bietet darüber hinaus weitere Lernverfahren an [Zel95]. Je nach Verlauf der Lernkurve können interaktiv Parameter des Lernverfahrens oder die Topologie des Netzes verändert werden, bis ein befriedigendes Ergebnis erzielt wird [Zel94].

### A.2.3 Aufbereitung des neuronalen Netzes

Mit Hilfe des dazu gehörigen Werkzeugs `snns2c` können die trainierten neuronalen Netze in C-Quelltext übersetzt werden. Der entstehende C-Code beinhaltet als Konstanten sämtliche Parameter des Netzes. Für jede Neuronenschicht finden sich dort Anweisungen, wie die folgenden:

```
for (member = 0; member < 10; member++) {
    unit = Hidden2[member];
    sum = 0.0;
    for (source = 0; source < unit->NoOfSources; source++) {
        sum += unit->sources[source]->act
            * unit->weights[source];
    }
    unit->act = Act_Logistic(sum, unit->Bias);
};
```

Damit ist es möglich, sie direkt in die Software zur Parametertransformation und -rücktransformation einzubinden. Durch das Compilieren in eine dynamische Bibliothek können diese Funktionen auch vom Java-Programm „TransHyb“ genutzt werden.

## A.3 Der Modellator „TransHyb“

Sowohl die Implementierung der vorgestellten Modelle als auch Experimente damit spielen im Rahmen dieser Arbeit eine wichtige Rolle. An dieser Stelle sollen kurz die wichtigsten Eigenschaften des Programms „TransHyb“, das dafür entwickelt wurde, skizziert werden (siehe auch [Mar03]).

Die folgenden Funktionen werden für die Arbeit mit hybriden Modellen bereitgestellt:

- Konstruktion der Struktur des Transportsystems: Platzieren und Verbinden der Transportsystem-Elemente, Festlegung der Parameter (Abbildung A.2).
- Festlegen von Pfaden durch das System. Jeder Pfad ist ein Gutstrom, der eine Folge von Transportsystem-Elementen passiert. Damit werden intern die lokalen Transportmatrizen erzeugt.
- Speichern und Laden von Systemen. Die Daten werden als XML-File abgelegt.
- Manuelle Festlegung der Berechnungsreihenfolge.
- Berechnung der statistischen Kenngrößen für die Komponenten.
- Zu jeder Zeit ist dabei die manuelle Eingabe von Variationskoeffizienten einzelner Ströme möglich.

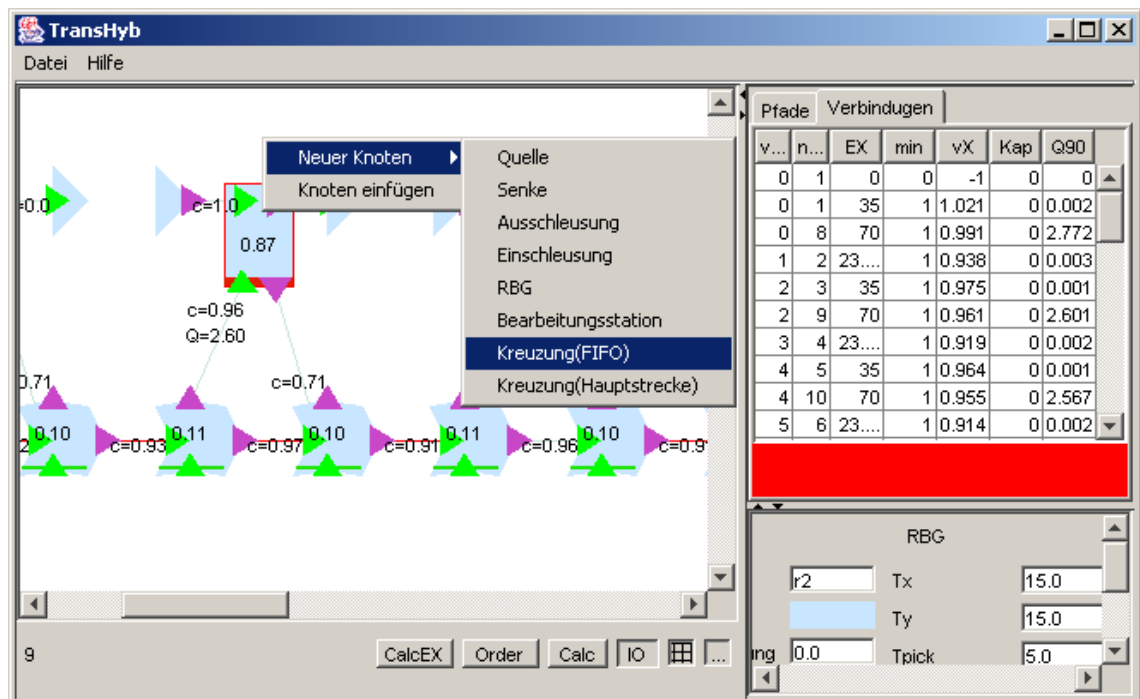


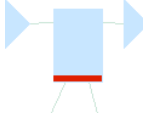






Abbildung A.2: TransHyb - Bedienoberfläche

Sämtliche modellierten Elemente werden als Polygone dargestellt. Die Ein- und Ausgänge der Transportsystem-Elemente können ein- und ausgeblendet werden. Verbindungen zwischen zwei Elementen werden als gerade Linien zwischen den entsprechenden Ein- und Ausgängen gezeichnet. Auch die Ergebnisse der Berechnungen können wahlweise ein- und ausgeblendet werden. Dazu gehören die Auslastungen

der Elemente, die Variationskoeffizienten und Quantile der Warteschlangen der Verbindungen zwischen den Komponenten.

Darstellung	Erklärung
	<b>Quelle.</b> Einzig möglicher Ausgangspunkt von Materialströmen
	<b>Senke.</b> Einzig möglicher Zielpunkt von Materialströmen
	<b>Regalbediengerät.</b> Auslagerungsaufträge müssen aus einer Quelle (links) zum Regal gebracht werden. Ausgang für Einlagerungen ist immer eine Senke (rechts).
	<b>Drehtisch (Vorfahrt).</b> Die Hauptstrecke verläuft von links nach rechts, die Nebenstrecke von unten nach oben.
	<b>Drehtisch (FCFS-Steuerung).</b> Gleichberechtigte Eingänge liegen links und unten.
	<b>Ein- oder Ausschleusung.</b> Ein- und Ausschleusungen sind in dieser Ansicht nicht unterscheidbar. Der Kontext im System läßt jedoch meist zweifelsfrei erkennen, um welches Transportsystem-Element es sich handelt.
	<b>Bedienstation.</b> Ein- und Ausgang der Bedienstation sind hier nicht unterscheidbar, die Zuordnung geht jedoch meist aus dem Kontext hervor.

**Tabelle A.1:** Legende der auftretenden Symbole

Tabelle A.1 zeigt die auftretenden Symbole. Das Einblenden der Ein- und Ausgänge der Transportsystem-Elemente ermöglicht in jedem Falle die zweifelsfreie Identifikation der Ein- und Ausgänge.

## A.4 Komplexe diskrete Simulationen

Für die Simulation komplexer Transportsysteme ist das Programm AutoMod (Brooks PRI) sehr gut geeignet. Es unterstützt sowohl die Entwicklung als auch den Lauf von Simulationsmodellen. Eine leistungsfähige Visualisierung ermöglicht frühzeitiges Erkennen von Modellierungsfehlern. Zahlreiche Funktionen gestatten einen Zugriff auf die Materialfluss-Steuerung des Modells.

AutoMod ist zur Entwicklung von großen Transportsystemen mit komplizierten Steuerungsregeln sehr gut geeignet. (Das belegen zum Beispiel die Simulationen für Infineon Dresden [Mar98b], [Mar00] oder für den UPS Worldport Louisville/Kentucky (USA) [Mar99].)

Auch wenn bei den Beispielen in Kapitel 5 die Steuerung nur grundlegende Aufgaben zu erfüllen hat, sind die Simulationen mit AutoMod gut zu realisieren.

## A.5 Mathematische Auswertung

Die Auswertung und Aufbereitung der Daten erfolgte mit Mathematica (Wolfram Research) [Wol93]. Dabei handelt es sich um ein Werkzeug für anspruchsvolle mathematische Analysen und Manipulationen. Mathematica ermöglicht:

- algebraische Manipulation von Ausdrücken,
- Integrieren (exakt/numerisch), Differenzieren,
- analytische oder numerische Lösung von Gleichungen,
- statistische Untersuchungen von Daten,
- Visualisieren von Funktionen und Daten.

Es stehen zahlreiche Module (packages) zur Verfügung u.a. für Statistik, lineare Algebra, Differential- und Integralrechnung.

# Anhang B

## Fehler der Modelle einzelner Elemente

An dieser Stelle finden sich Fehlerdiagramme, die zwar ihren jeweiligen Beitrag zur Verifikation und Validierung leisten, aber nicht explizit in Kapitel 4 diskutiert werden. Für jedes der betrachteten Transportsystem-Elemente wurden die interessierenden Größen (90%-Quantil der Warteschlangen an den Eingängen, Auslastung des Elements und die Variationskoeffizienten der Abgangsverteilung) bezüglich ihrer Parameter einzeln mit neuronalen Netzen approximiert.

Für jeden Element-Typ und jede approximierte Größe werden hier die gleichen Experimente veranschaulicht:

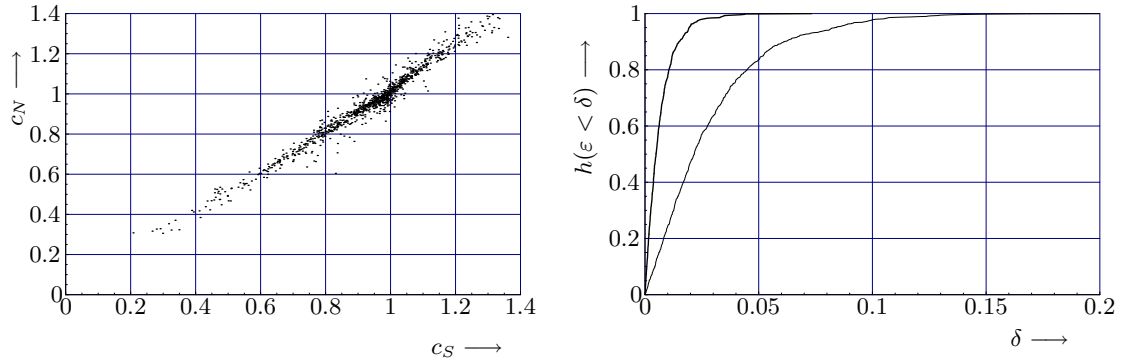
1000 Tupel wurden zufällig (in jeder Dimension gleichverteilt) aus dem gesamten Parameterraum ausgewählt. Jedem dieser Tupel entspricht ein konkretes Transportsystem-Element mit definierten logistischen Parametern (Erwartungswert und Variationskoeffizient der Eingangsströme, lokale Transportmatrix). Für dieses Element wurde eine diskrete Simulation mit zweipunktverteilten Eingangsströmen durchgeführt. Während die neuronalen Netze jeweils mit gammaverteilten Eingangsströmen trainiert wurden, sind hier also beträchtliche Fehler zu erwarten.

In den jeweils linken Diagrammen sind die Netzprognosen über den zugehörigen Simulationsdaten aufgetragen. In den rechten Diagrammen werden die empirischen Verteilungen der Fehler dargestellt (dünnere Kurve). Diese Diagramme dienen der Validierung der einzelnen Modelle.

Darüberhinaus wurden weitere 1000 Experimente mit gammaverteilten Eingangsströmen durchgeführt (Verifikation). Das heißt, es handelt sich hier um Probleme, die durch die neuronalen Netze sehr genau approximiert werden sollten, da sie für Gammaverteilungen trainiert wurden. Die Verteilung dieser Fehler wird zum Vergleich ebenfalls in den jeweils rechten Diagrammen dargestellt (fettgedruckte Kurve). In jedem Fall ist der Fehler hierbei deutlich kleiner als bei den Experimenten mit zweipunktverteilten Eingangsströmen.

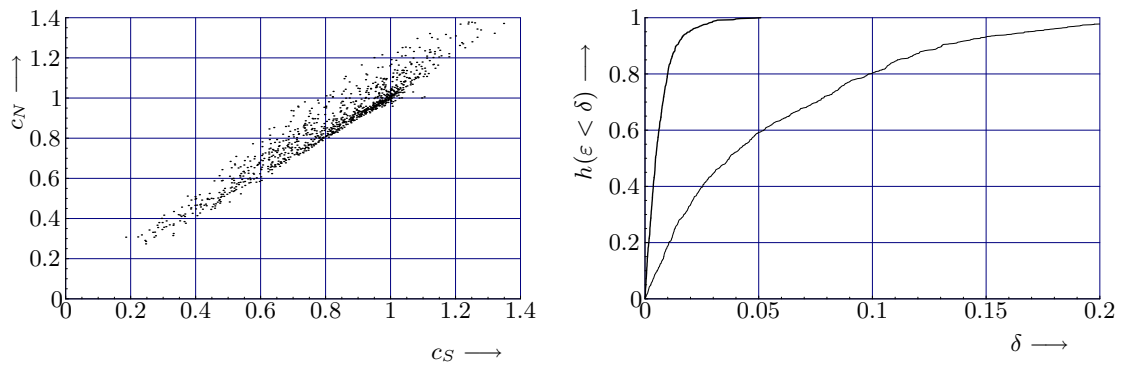
## B.1 Ausschleusung

### B.1.1 Variationskoeffizient des Ausgangs der Hauptstrecke



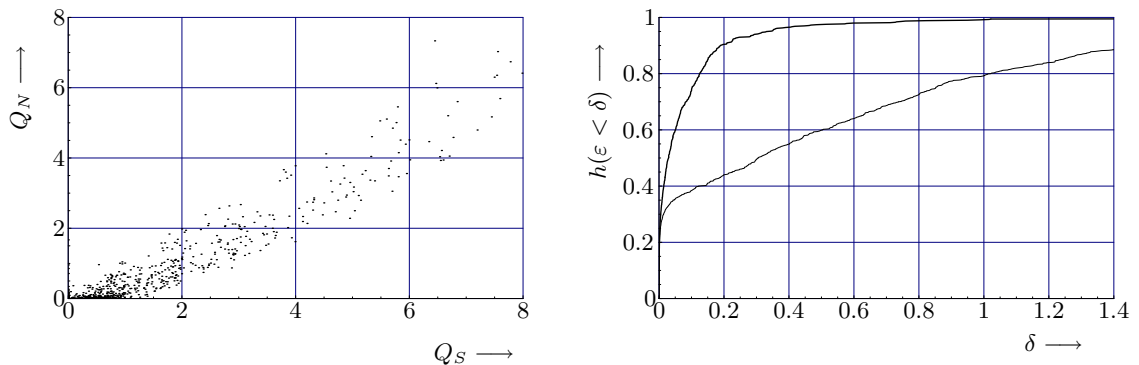
**Abbildung B.1:** Links: Vergleich der Simulationsdaten  $c_S$  mit den Netzprognosen  $c_N$ . Rechts: Verteilung des Fehlers  $\varepsilon$ . Fette Kurve: gammadistribuierte Eingangsstrom; dünne Kurve: zweipunktdistribuierte Eingangsstrom.

### B.1.2 Variationskoeffizient des Ausgangs der Nebestrecke



**Abbildung B.2:** Links: Vergleich der Simulationsdaten  $c_S$  mit den Netzprognosen  $c_N$ . Rechts: Verteilung des Fehlers  $\varepsilon$ . Fette Kurve: gammadistribuierte Eingangsstrom; dünne Kurve: zweipunktdistribuierte Eingangsstrom.

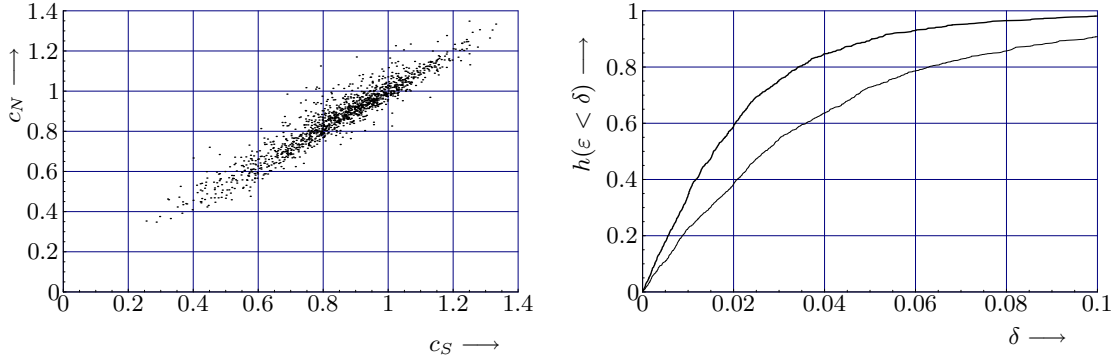
### B.1.3 90%-Quantil der Warteschlange



**Abbildung B.3:** Links: Vergleich der Simulationsdaten  $Q_S$  mit den Netzprognosen  $Q_N$ . Rechts: Verteilung des Fehlers  $\varepsilon$ . Fette Kurve: gammadistribuierte Eingangsstrom; dünne Kurve: zweipunktdistribuierte Eingangsstrom.

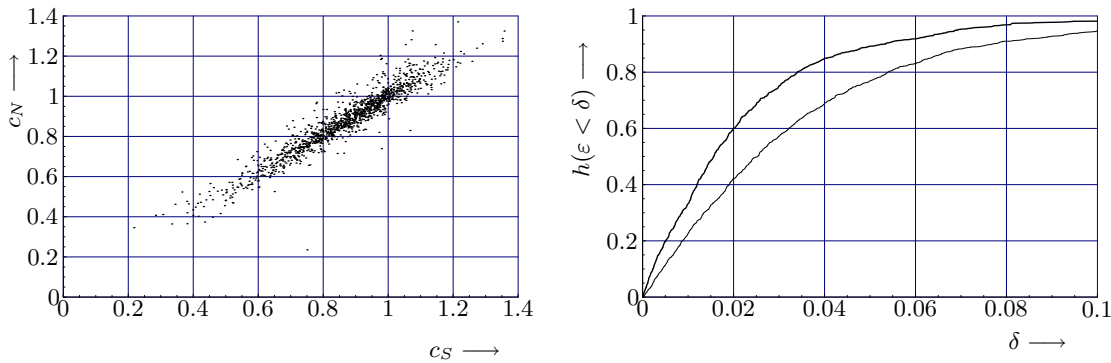
## B.2 Kreuzung mit FCFS-Steuerung

### B.2.1 Variationskoeffizient des Ausgangs 3 (Hauptstrecke)



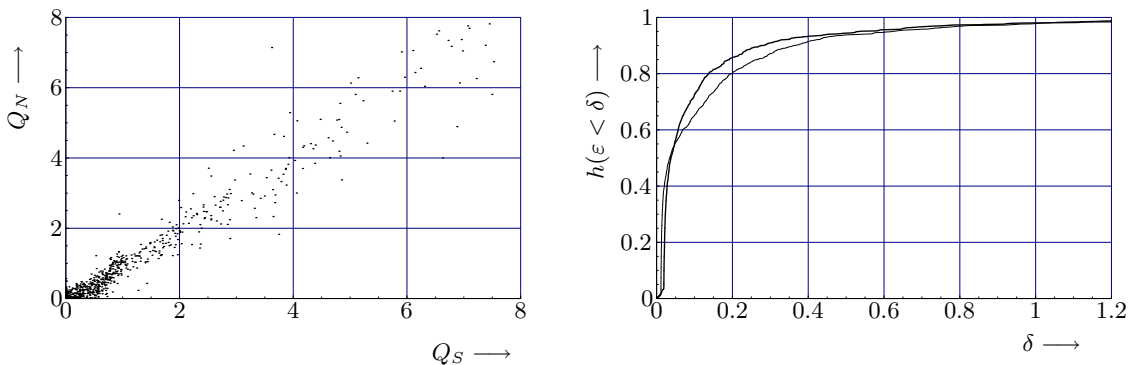
**Abbildung B.4:** Links: Vergleich der Simulationsdaten  $c_S$  mit den Netzprognosen  $c_N$ . Rechts: Verteilung des Fehlers  $\varepsilon$ . Fette Kurve: gammaverteilter Eingangsstrom; dünne Kurve: zweipunktverteilter Eingangsstrom.

### B.2.2 Variationskoeffizient des Ausgangs 4 (Nebenstrecke)



**Abbildung B.5:** Links: Vergleich der Simulationsdaten  $c_S$  mit den Netzprognosen  $c_N$ . Rechts: Verteilung des Fehlers  $\varepsilon$ . Fette Kurve: gammaverteilter Eingangsstrom; dünne Kurve: zweipunktverteilter Eingangsstrom.

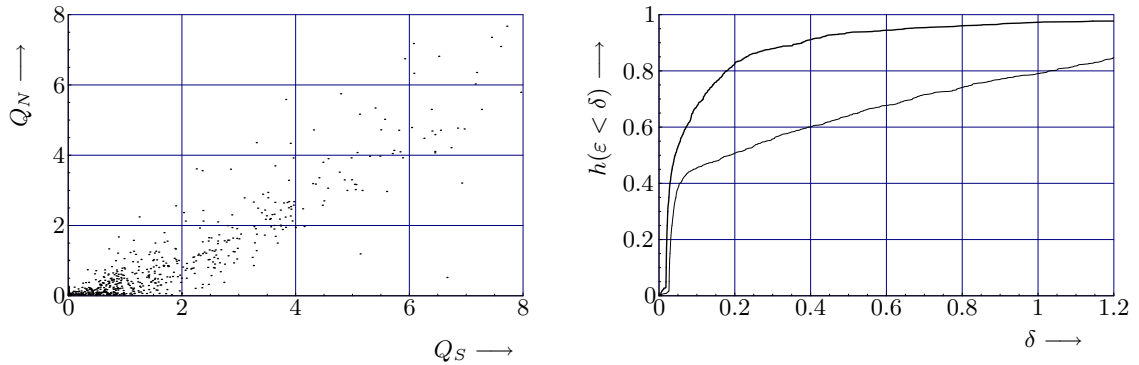
### B.2.3 90%-Quantil der Warteschlange (Hauptstrecke)



**Abbildung B.6:** Links: Vergleich der Simulationsdaten  $Q_S$  mit den Netzprognosen  $Q_N$ . Rechts: Verteilung des Fehlers  $\varepsilon$ . Fette Kurve: gammaverteilter Eingangsstrom; dünne Kurve: zweipunktverteilter Eingangsstrom.

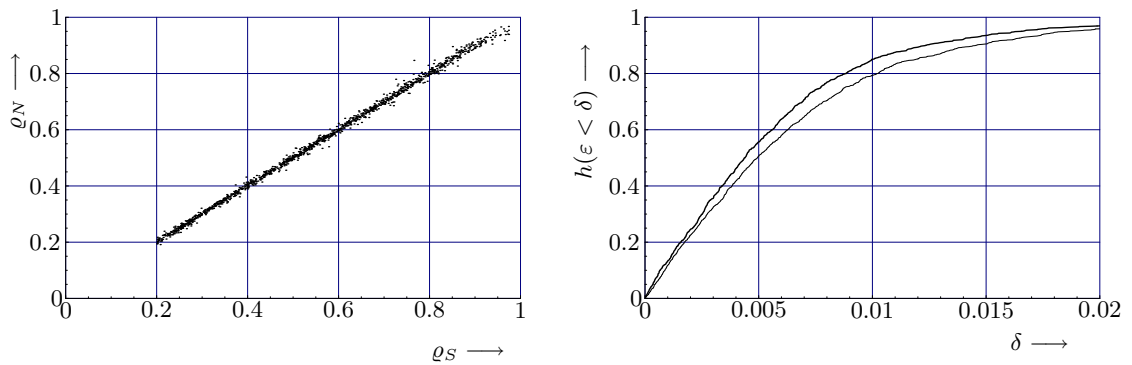


### B.2.4 90%-Quantil der Warteschlange (Nebenstrecke)



**Abbildung B.7:** Links: Vergleich der Simulationsdaten  $Q_S$  mit den Netzprognosen  $Q_N$ . Rechts: Verteilung des Fehlers  $\varepsilon$ . Fette Kurve: gammaverteilter Eingangsstrom; dünne Kurve: zweipunktverteilter Eingangsstrom.

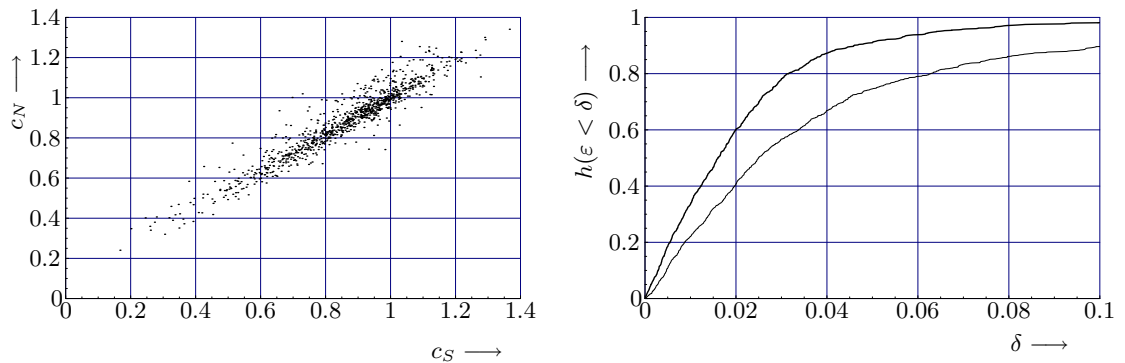
### B.2.5 Auslastung



**Abbildung B.8:** Links: Vergleich der Simulationsdaten  $\rho_S$  mit den Netzprognosen  $\rho_N$ . Rechts: Verteilung des Fehlers  $\varepsilon$ . Fette Kurve: gammaverteilter Eingangsstrom; dünne Kurve: zweipunktverteilter Eingangsstrom.

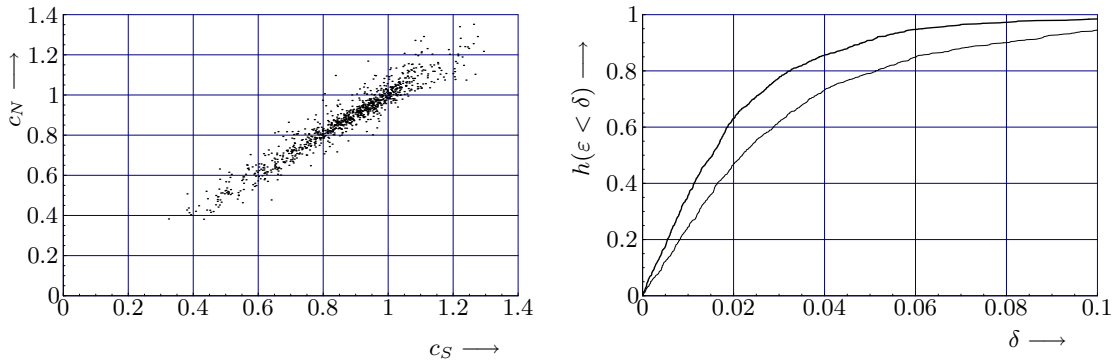
## B.3 Kreuzung mit Vorfahrtsregelung

### B.3.1 Variationskoeffizient des Ausgangs 3 (Hauptstrecke)



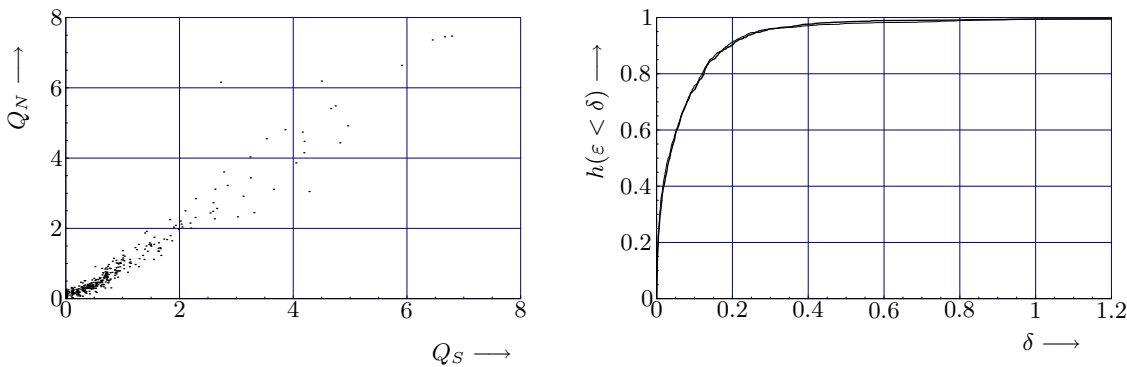
**Abbildung B.9:** Links: Vergleich der Simulationsdaten  $c_S$  mit den Netzprognosen  $c_N$ . Rechts: Verteilung des Fehlers  $\varepsilon$ . Fette Kurve: gammaverteilter Eingangsstrom; dünne Kurve: zweipunktverteilter Eingangsstrom.

### B.3.2 Variationskoeffizient des Ausgangs 4 (Nebenstrecke)



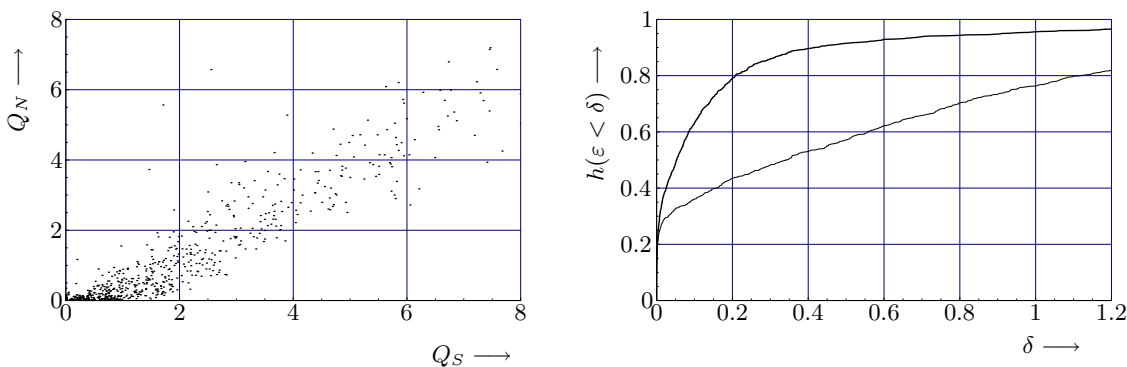
**Abbildung B.10:** Links: Vergleich der Simulationsdaten  $c_S$  mit den Netzprognosen  $c_N$ . Rechts: Verteilung des Fehlers  $\varepsilon$ . Fette Kurve: gammaverteilter Eingangsstrom; dünne Kurve: zweipunktverteilter Eingangsstrom.

### B.3.3 90%-Quantil der Warteschlange (Hauptstrecke)



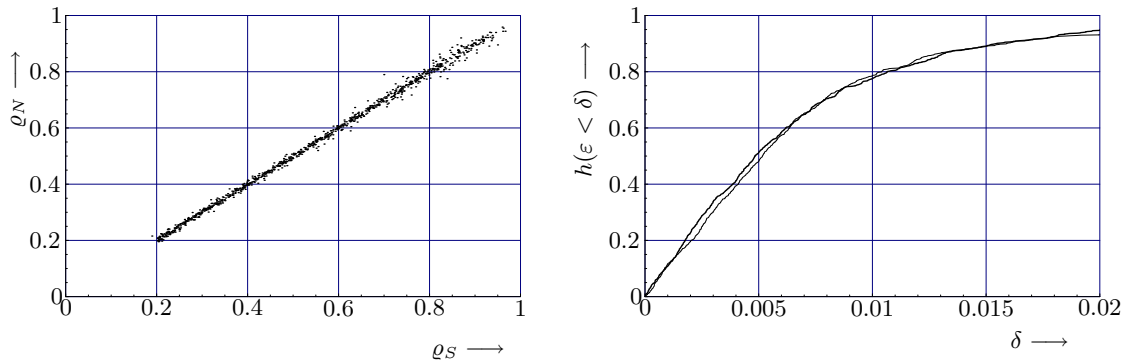
**Abbildung B.11:** Links: Vergleich der Simulationsdaten  $Q_S$  mit den Netzprognosen  $Q_N$ . Rechts: Verteilung des Fehlers  $\varepsilon$ . Fette Kurve: gammaverteilter Eingangsstrom; dünne Kurve: zweipunktverteilter Eingangsstrom.

### B.3.4 90%-Quantil der Warteschlange (Nebenstrecke)



**Abbildung B.12:** Links: Vergleich der Simulationsdaten  $Q_S$  mit den Netzprognosen  $Q_N$ . Rechts: Verteilung des Fehlers  $\varepsilon$ . Fette Kurve: gammaverteilter Eingangsstrom; dünne Kurve: zweipunktverteilter Eingangsstrom.

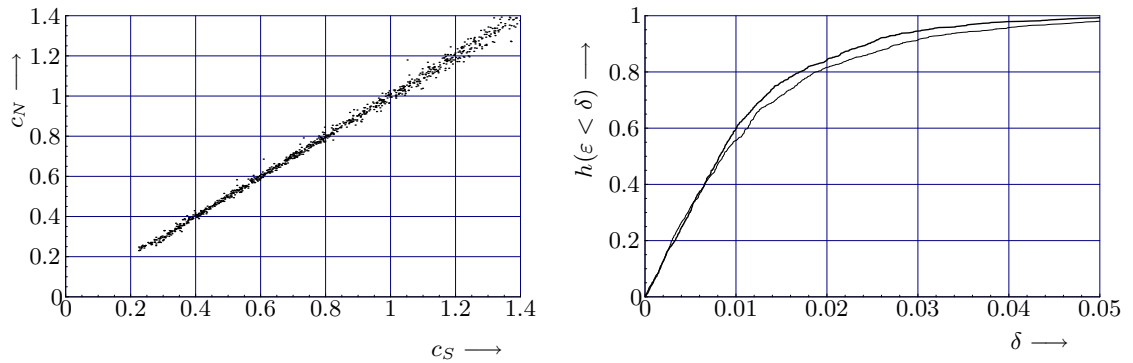
### B.3.5 Auslastung



**Abbildung B.13:** Links: Vergleich der Simulationsdaten  $\rho_S$  mit den Netzprognosen  $\rho_N$ . Rechts: Verteilung des Fehlers  $\varepsilon$ . Fette Kurve: gammaverteilter Eingangsstrom; dünne Kurve: zweipunktverteilter Eingangsstrom.

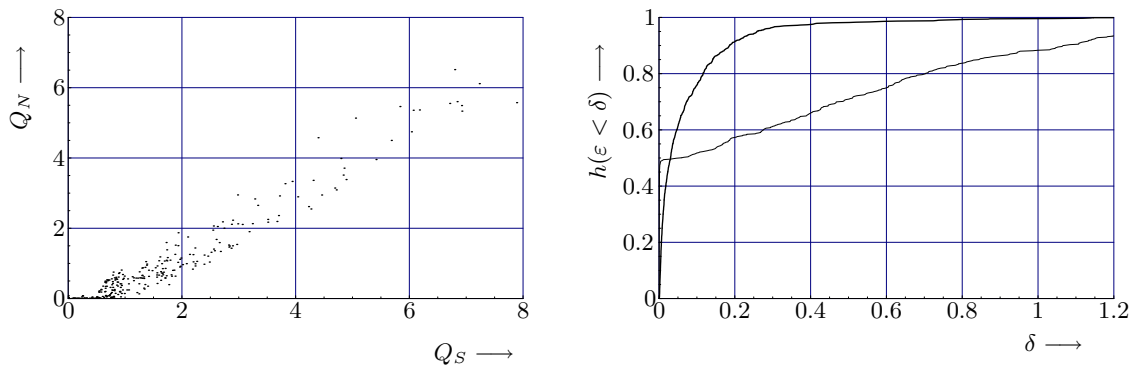
## B.4 Regalbediengerät

### B.4.1 Variationskoeffizient des Ausgangs



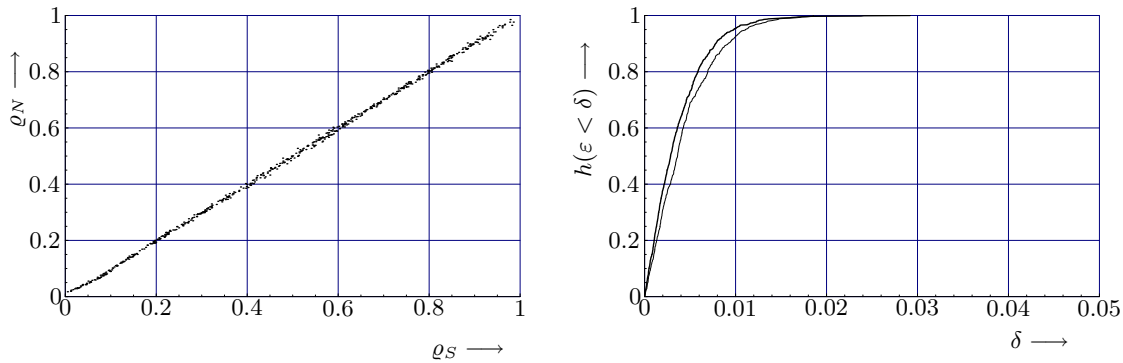
**Abbildung B.14:** Links: Vergleich der Simulationsdaten  $c_S$  mit den Netzprognosen  $c_N$ . Rechts: Verteilung des Fehlers  $\varepsilon$ . Fette Kurve: gammaverteilter Eingangsstrom; dünne Kurve: zweipunktverteilter Eingangsstrom.

### B.4.2 90%-Quantil der Warteschlange vor dem Lager



**Abbildung B.15:** Links: Vergleich der Simulationsdaten  $Q_S$  mit den Netzprognosen  $Q_N$ . Rechts: Verteilung des Fehlers  $\varepsilon$ . Fette Kurve: gammaverteilter Eingangsstrom; dünne Kurve: zweipunktverteilter Eingangsstrom.

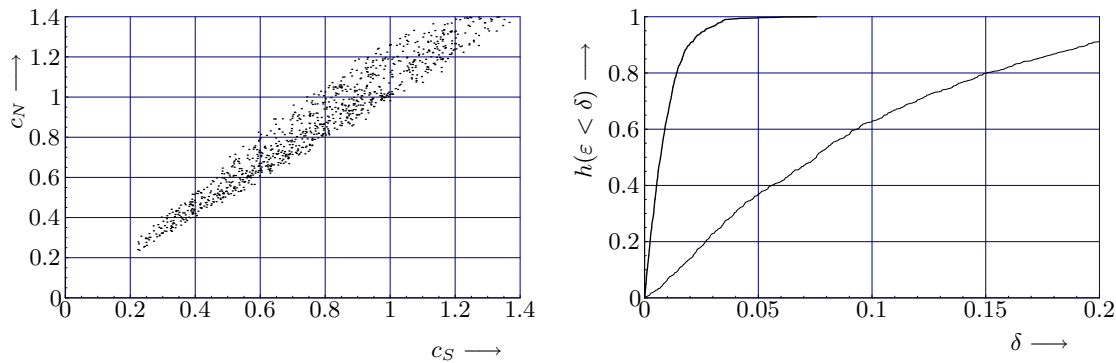
### B.4.3 Auslastung des Regalbediengeräts



**Abbildung B.16:** Links: Vergleich der Simulationsdaten  $q_S$  mit den Netzprognosen  $q_N$ . Rechts: Verteilung des Fehlers  $\varepsilon$ . Fette Kurve: gammadaverteilter Eingangsstrom; dünne Kurve: zweipunktverteilter Eingangsstrom.

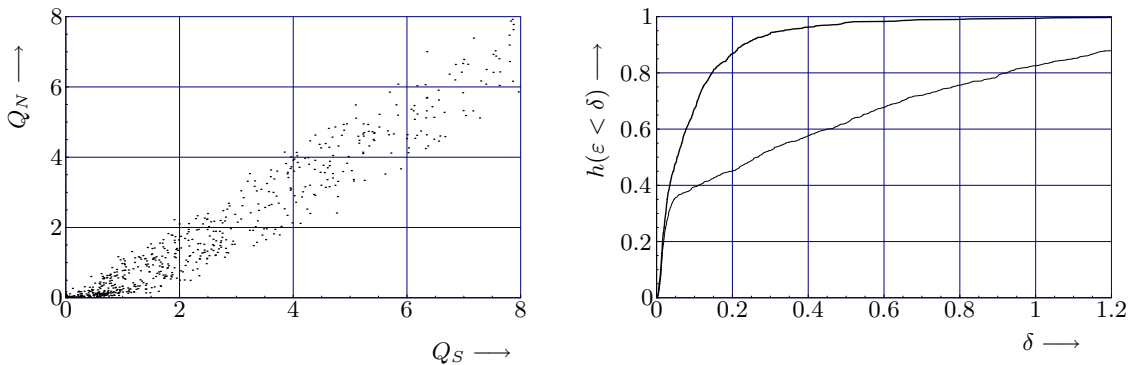
## B.5 Bedienstation

### B.5.1 Variationskoeffizient des Ausgangs



**Abbildung B.17:** Links: Vergleich der Simulationsdaten  $c_S$  mit den Netzprognosen  $c_N$ . Rechts: Verteilung des Fehlers  $\varepsilon$ . Fette Kurve: gammadaverteilter Eingangsstrom; dünne Kurve: zweipunktverteilter Eingangsstrom.

### B.5.2 90%-Quantil der Warteschlange



**Abbildung B.18:** Links: Vergleich der Simulationsdaten  $Q_S$  mit den Netzprognosen  $Q_N$ . Rechts: Verteilung des Fehlers  $\varepsilon$ . Fette Kurve: gammadaverteilter Eingangsstrom; dünne Kurve: zweipunktverteilter Eingangsstrom.

# Anhang C

## Verzeichnis verwendeter Symbole

Mit Großbuchstaben  $X, Y, Z$  werden Zufallsgrößen oder Ströme bezeichnet.  $X_i$  sind dabei vorrangig Eingangsströme,  $Y_j$  Ausgangsströme.  $Z$  wird meist benutzt, wenn es unerheblich ist, ob sich die Aussage auf einen Eingang oder Ausgang bezieht.

$\mathcal{A}(o) \dots$  Attribute eines Objektes  $o$

$b \dots$  Regalwandparameter

$c, c_Z \dots$  Variationskoeffizient, Variationskoeffizient einer Zufallsgröße oder eines Stroms  $Z$

$D^2 Z, D^2(Z) \dots$  Streuung einer Zufallsgröße oder eines Stroms  $Z$

$e \dots$  Eulersche Konstante

$EZ, E(Z) \dots$  Erwartungswert einer Zufallsgröße oder eines Stroms  $Z$

$f_Z(\cdot) \dots$  Dichtefunktion einer Zufallsgröße  $Z$

$F(\cdot), F_Z(\cdot) \dots$  statistische Verteilungsfunktion, Verteilungsfunktion einer Zufallsgröße  $Z$

$h(\cdot) \dots$  beobachtete relative Häufigkeit

$M \dots$  Transportmatrix eines Systems

$\mathcal{M} \dots$  lokale Transportmatrix

$\mathbf{M} \dots$  Modellierung (Abbildung zwischen zwei Systemen)

$M_t \dots$  Daten für das Training eines neuronalen Netzes

$n_{in}, n_{out} \dots$  Anzahl der Ein- bzw. Ausgänge eines Transportsystem-Elements

$\mathbf{net}(\cdot) \dots$  Berechnungsfunktion eines neuronalen Netzes (Ausgabevektor)

$\mathbb{N} \dots$  Menge der natürlichen Zahlen

- $\mathcal{O} \dots$  Menge von Objekten
- $P(\cdot) \dots$  Wahrscheinlichkeit für ein Ereignis
- $Q_p(X) \dots$  Quantil der Ordnung  $p$  einer Zufallsgröße  $X$
- $Q_{90}(X), Q(X) \dots$  Quantil der Ordnung 90% der Länge der Warteschlange am Eingang  $X$
- $\mathcal{R} \dots$  Menge von Beziehungen zwischen Objekten
- $\mathbb{R} \dots$  Menge der reellen Zahlen
- $\mathcal{S} \dots$  System
- $s_{min} \dots$  minimaler Abstand zwischen zwei eintreffenden Transporteinheiten eines Gutstroms
- $\mathcal{T} \dots$  Indexmenge der Systemzeit
- $t_b \dots$  Bedienzeit
- $t_{min}, t_{min}(Z) \dots$  minimale Zeit zwischen zwei eintreffenden Forderungen oder Transporteinheiten eines Stroms  $Z$
- $v \dots$  Transportgeschwindigkeit eines Förderers
- $w \dots$  Gewichtungsfaktor in neuronalem Netz
- $x \dots$  Eingang eines Neurons
- $y \dots$  Ausgang eines Neurons
- $z \dots$  gewichtete Summe eines Neurons
- $\mathcal{Z}, \mathcal{Z}_t \dots$  Systemzustand, Systemzustand zur Zeit  $t$
- $\varepsilon \dots$  Fehler
- $\boldsymbol{\pi} = (\pi_1, \pi_2, \dots) \dots$  Parametervektor
- $\boldsymbol{\pi}(Z) \dots$  Parameterdarstellung eines Stromes  $Z$
- $\Pi \dots$  Menge von Parametervektoren (Stützstellen)
- $\mathbb{R} \dots$  Menge der reellen Zahlen
- $\varrho \dots$  Auslastung
- $\sigma_{b,c}(\cdot) \dots$  logistische Funktion
- $\tau \dots$  Zeit eines Ereignisses
- $\tau_E \dots$  Eingangereignis

$\tau_A \dots$  Abgangseignis

$\psi(\cdot) \dots$  Aktivierungsfunktion

$\chi(\mathcal{A}), \chi_t(\mathcal{A}) \dots$  Wertebelegung der Attribute  $\mathcal{A}$ , Wertebelegung zur Zeit  $t$

# Literaturverzeichnis

- [Alt87] ALTIOK, T.: *Approximate analysis of arbitrary configurations of open queueing networks with blocking*. Annals of Operations Research, 9: S. 481–509, 1987.
- [Ant99] ANTONIADIS, A.: *Wavelets in statistics: a review*. Journal of the Italian Statistical Society, 6(2): S. 97–144, 1999.
- [Arn02] ARNOLD, D. (Herausgeber): *Handbuch Logistik*. Springer, 2002.
- [Bab97] BABEL, W.: *Einsatzmöglichkeiten neuronaler Netze in der Industrie*. Expert Verlag, Renningen-Malmsheim, 1997.
- [Ban98] BANKS, J.: *Handbook of simulation : principles, methodology, advances, applications, and practice*. Wiley, New York, 1998.
- [Ban01] BANKS, J.: *Discrete event simulation*. Prentice Hall, Upper Saddle River, NJ, 2001.
- [Bie95] BIELE, C.: *COSMO: Entwurf und Realisierung einer modifizierten DESMO-Komponente zur diskret-kontinuierlichen Simulation*. Diplomarbeit, Universität Hamburg, 1995.
- [Bol89] BOLCH, G.: *Leistungsbewertung von Rechensystemen mittels analytischer Warteschlangenmodelle*. Teubner, Stuttgart, 1989.
- [Bor94] BORRISS, MARTIN: *Simulation von Warteschlangennetzen in BNETD*. Diplomarbeit, TU Dresden, Fak. Informatik, Inst. f. Systemarchitektur, 1994.
- [Bos94] BOSSEL, H.: *Modellbildung und Simulation*. Vieweg, Braunschweig, Wiesbaden, 1994.
- [Bra97] BRAUN, H., FEULNER, J., MALAKA, R.: *Praktikum Neuronale Netze*. Springer, Berlin, Heidelberg, New York, 1997.
- [Bro91] BROOKS, R.A.: *Intelligence without reason*. In: *Twelfth International Joint Conference on Artificial Intelligence*, Sidney, 1991. AAAI Press.
- [Bur00] BURGER, M., ENGL, H.W.: *Training neural networks with noisy data as an ill-posed problem*. Advances in Computational Mathematics, 13(4): S. 335–354, 2000.



- [Car86] CARSON, J.: *Convincing users of model's validity is challenging aspect of modeller's job*. Industrial Engineering, 18 (6): S. 74–85, 1986.
- [Cha95] CHANG, G.L., SU, C. C.: *Predicting intersection queue with neural network models*. Transportation Research, 3(3): S. 175–191, 1995.
- [Chi96] CHIU, C.-T., MEHROTRA, K., MOHAN, C.K., RANKA, S.: *Robustness of feedforward neural networks*. In: SIMPSON, P. (Herausgeber): *Neural networks theory, technology, and applications*, S. 248–353. IEEE, New York, 1996.
- [Cyb89] CYBENKO, G.: *Approximation by superposition of sigmoidal functions*. Mathematics of Control, Signals, and Systems, 2, Nr.4: S. 303–314, 1989.
- [Fer64] FERSCHL, F.: *Zufallsabhängige Wirtschaftsprozesse*. Physica-Verlag, Wien-Würzburg, 1964.
- [Fis76] FISZ, M.: *Wahrscheinlichkeitsrechnung und mathematische Statistik*. VEB Deutscher Verlag der Wissenschaften, Berlin, 1976.
- [For70] FORRESTER, J.W.: *World dynamics*. MIT Press, Cambridge, 1970.
- [Fur92] FURMANS, K.: *Ein Beitrag zur theoretischen Behandlung von Materialflußpuffern in Bediensystemnetzwerken*. Dissertation, Universität Karlsruhe, 1992.
- [Gra95] GRAPS, A.: *An introduction to wavelets*. IEEE Computational Science and Engineering, 2(2): S. 50–61, 1995.
- [Gra97] GRAUPE, D.: *Principles of Artificial Neural Networks*. World Scientific, Singapore, 1997.
- [Gri00] GRIEWANK, A.: *Evaluating derivatives : principles and techniques of algorithmic differentiation*. Society for Industrial and Applied Mathematics, Philadelphia, 2000.
- [Gro84] GROSSESCHALLAU, W.: *Materialflußrechnung*. Springer, 1984.
- [Gud75a] GUDEHUS, T.: *Grenzleistungen bei absoluter Vorfahrt*. Zeitschrift für Operations Research, 20: S. B127–B160, 1975.
- [Gud75b] GUDEHUS, T.: *Staueffekte vor Transportknoten*. Zeitschrift für Operations Research, 20: S. B207–B252, 1975.
- [Gud99] GUDEHUS, T.: *Logistik*. Springer, 1999.
- [Heb49] HEBB, D.: *The organization of behavior*. Wiley, New York, 1949.
- [Heu92] HEUSER, H.: *Funktionalanalysis*, Band 2. Teubner, Stuttgart, 1992.

- [Hop85] HOPFIELD, J.J., TANK, D.W.: *Neural computations of decisions in optimization problems*. Biological Cybernetics, 52: S. 141–152, 1985.
- [Hun99] HUNG, Y.F., LEACHMAN, R.C.: *Reduced simulation models of wafer fabrication facilities*. International Journal of Production Research, 37 (12): S. 2685–2701, 1999.
- [Iov00] IOVINE, J.: *Understanding Neural Networks*. Prompt, Indianapolis, 2000.
- [Jün00] JÜNEMANN, R., SCHMIDT, T.: *Materialflußsysteme, Systemtechnische Grundlagen*. Springer, 2000.
- [Kan98] KANSTEIN, A.: *Possibilistic neural networks for process modelling*. VDE-Verlag, 1998.
- [Kee99] KEEL, A.: *Beschreibende Statistik*. Verlag Wilhelm Surbir, St. Gallen, 1999.
- [Koh72] KOHONEN, T.: *Correlation matrix memories*. IEEE Transactions on Computers, 21: S. 353–359, 1972.
- [Kra90] KRATZER, K. P.: *Neuronale Netze*. Hanser-Verlag, München, Wien, 1990.
- [Kre01] KREPPENHOFER, L., MARQUARDT, H.-G., MARKWARDT, U., GLASS, M.: *Regalbediengeräte mit Mehrfach-Lastaufnahme: Planungsrichtlinie und Steuerungsverfahren*. Forschungsbericht FE 148, TU Dresden, Institut für Fördertechnik, Baumaschinen und Logistik, 2001.
- [Kru99] KRUG, W., LIEBELT, J., BAUMBACH, B.: *Intelligentes System eines automatisch gesteuerten simulationsbasierten Optimierung*. In: KÖCHEL, P. (Herausgeber): *13. Workshop der ASIM-Fachgruppe 4.5.3. „Simulation und Künstliche Intelligenz“*, KI-Methoden in der simulationsbasierten Optimierung, TU Chemnitz, 1999. TU Chemnitz, Fak. Informatik.
- [Kur01] KURKOVA, V., SANGUINETI, M.: *Tight bounds on rates of neural-network approximation*. In: DORFFNER, G. ET AL (Herausgeber): *International Conference on Artificial Neural Networks (ICANN)*, S. 277–282, Wien, 2001. Springer.
- [Law91] LAW, A.M., KELTON, W.D.: *Simulation modeling and analysis*. McGraw-Hill Inc., 1991.
- [Lec99] LECHLER, T.: *Entwurf und Implementierung eines Frameworks für diskrete Simulatoren in Java*. Diplomarbeit, Universität Hamburg, FB Informatik, 1999.
- [Lo01] LO, J.T.: *Robust neural networks*. In: SANDBERG, I.W. (Herausgeber): *Nonlinear dynamical systems - feedforward neural network perspectives*. Wiley, New York, 2001.

- [Mai00] MAIOROV, V.E., MAIR, R.: *On the near optimality of the stochastic approximation of smooth functions by neural networks*. Advances in Computational Mathematics, 13(1): S. 79–103, 2000.
- [Mar98a] MARKWARDT, U.: *Erarbeitung einer Planungs- und Projektierungsrichtlinie und Erarbeitung von Regeln zum Leistungsnachweis für Horizontalkarussellager*. Forschungsbericht FE 46, TU Dresden, Institut für Fördertechnik, Baumaschinen und Logistik, 1998.
- [Mar98b] MARKWARDT, U.: *Simulation der Abläufe im Transportsystem und in der Fertigungslogistik der Waferproduktion*. Forschungsbericht FE 97, TU Dresden, Institut für Fördertechnik, Baumaschinen und Logistik, 1998.
- [Mar99] MARKWARDT, U., GLASS, M.: *Simulation HUB 2000 - AutoMod model for the irregular package transport system at the UPS facility in Louisville (USA)*. Forschungsbericht FE 107, TU Dresden, Institut für Fördertechnik, Baumaschinen und Logistik, 1999.
- [Mar00] MARQUARDT, H.-G., MARKWARDT, U.: *Simulation of a conveyor-based transportation system in a semiconductor factory*. In: *SimCon Symposium*, Gent, 2000.
- [Mar03] MARKWARDT, U.: *Prototypische Implementierung eines hybriden Modells zur Modellierung von Materialfluss-Systemen mit Hilfe von neuronalen Netzen*. Forschungsbericht FE 165, TU Dresden, Institut für Fördertechnik, Baumaschinen und Logistik, 2003.
- [McC43] MCCULLOCH, W.S., PITTS, W.: *A logical calculus of the ideas immanent in nervous activity*. Bulletin of Mathematical Biophysics, 5: S. 115–133, 1943. Anderson, J.A., Rosenfeld, E. (Hrsg.): *Neurocomputation: Foundations of research*. MIT-Press 1988.
- [Mha96] MHASKAR, H.N.: *Neural networks for localized approximation of real functions*. In: SIMPSON, PATRICK K. (Herausgeber): *Neural networks theory, technology, and applications*. Institute of Electrical and Electronics Engineers, New York, 1996.
- [Mha97] MHASKAR, H.N., HAHM, N.: *Neural networks for functional approximation and system identification*. Neural Computation, 9(1): S. 143–159, 1997.
- [Mha98] MHASKAR, H.N.: *Approximation of smooth functions by neural networks*. In: AL., K. WARWICK ET (Herausgeber): *Dealing with complexity: A neural network approach*, Perspectives in Neural Computing, S. 189–204. Springer, London, 1998.
- [Min85] MINSKY, M.: *The society of mind*. Simon & Schuster, New York, 1985.

- [Möl92] MÖLLER, D.F.P. (Herausgeber): *Modellbildung, Simulation und Identifikation dynamischer Systeme*. Springer, Berlin, Heidelberg, New York, 1992.
- [Moo94] MOODY, J.: *Prediction risk and architecture selection for neural networks*. In: CHERKASSKY, V. ET AL (Herausgeber): *From statistics to networks: Theory and pattern recognition applications*, NATO ASI Series F. Springer, 1994.
- [Nas99] NASEREDDIN, M., MOLLAGHASEMI, M.: *The development of a methodology for the use of neural networks and simulation modeling in system design*. In: P. A. FARRINGTON, H. B. NEMBHARD, D. T. STURROCK G. W. EVANS (Herausgeber): *Winter Simulation Conference*, Phoenix, Arizona, 1999.
- [Ost01] OSTERMANN, A.: *Neue Ansätze zur gesamtheitlichen Fabriksimulation*. Innovationen der Fabrikplanung und -organisation. Shaker Verlag, Clausthal, 2001.
- [Pag91] PAGE, B.: *Diskrete Simulation*. Springer, Berlin, Heidelberg, New York, 1991.
- [Pan00] PANAYIOTOU, C.G., CASSANDRAS, C.G., GONG, W.-B.: *Model abstraction for discrete event systems using neural networks and sensitivity information*. In: JOINES, J.A. ET AL (Herausgeber): *Proceedings of the 2000 Winter Simulation Conference*. <http://www.informs-cs.org/wsc00papers/048.PDF>, 2000.
- [Pen89] PENROSE, R.: *The emperor's new mind*. Vintage, London, 1989.
- [Qia98] QIAU, F., YANG, H.: *Neural network approach to classification of traffic flow states*. Journal of Transportation Engineering, 124: S. 521–525, 1998.
- [Qia01] QIAU, F., YANG, H., LAM, W.: *Intelligent simulation and prediction of traffic flow dispersion*. Transportation Research, B 35, 2001.
- [Ral98] RALL, B.: *Analyse und Dimensionierung von Materialflußsystemen mittels geschlossener Warteschlangennetze*. Dissertation, Universität Karlsruhe, 1998.
- [Rao96] RAO, S., PAPPU, R.: *neural networks for localized approximation of real functions*. In: SIMPSON, P. (Herausgeber): *Neural networks: theory technology, and applications*. IEEE, New York, 1996.
- [Ras76] RASCH, D.: *Einführung in die mathematische Statistik*, Band 2. VEB Deutscher Verlag der Wissenschaften, Berlin, 1976.
- [Rau88] RAUCH, H.E., WINARSKE, T.: *Neural networks for routing communication traffic*. IEEE Control Systems Magazine, 8(2): S. 26–31, 1988.

- [Rob97] ROBINSON, S.: *Simulation model verification and validation: increasing the users' confidence*. In: ANDRADÓTTIR, S. ET. AL. (Herausgeber): *Winter Simulation Conference*, S. 53–59, Atlanta (USA), 1997.
- [Roj93] ROJAS, R.: *Theorie der neuronalen Netze*. Springer, Berlin, Heidelberg, New York, 1993.
- [Ros62] ROSENBLATT, D.E. HINTON.: *A comparison of several perceptron models*. In: YOUTS, M.C. ET AL (Herausgeber): *Self organizing systems*. Spartan Books, Washington, 1962.
- [Rum86] RUMELHART, D.E., HINTON, G.E., WILLIAMS, R.J: *Learning internal representations by error propagation*. In: RUMELHART, D.E., MCCLELLAND, J.L. (Herausgeber): *Parallel distributed processing: explorations in the microstructure of cognition*, Band 1. The MIT Press, Cambridge, 1986.
- [Sab02] SABUNCUOGLU, I., TOUHAMI, S.: *Simulation metamodeling with neural networks: an experimental investigation*. International Journal of Production Research, 40(11): S. 2483–2505, 2002.
- [Sch70] SCHMIDT, J.W., TAYLOR, R.E.: *Simulation and analysis of industrial systems*. Homewood, Illinois, 1970.
- [Sch97] SCHERER, A.: *Neuronale Netze*. Vieweg, Braunschweig/Wiesbaden, 1997.
- [Sch99] SCHMIDT, T.: *Grobplanung mit künstlichen neuronalen Netzen für die Halbleiterfertigung*. Jost Jetter Verlag, Heimsheim, 1999.
- [Sch00a] SCHULZE, F.: *Neue Methoden zur Modellbildung für die Planung und Steuerung von Materialflusssystemen*. Dissertation, TU Dresden, 2000.
- [Sch00b] SCHULZE, F., MARQUARDT, H.-G.: *Neuronale Netze in der Materialflusssimulation*. In: MERTINS, K. (Herausgeber): *The new simulation in production and logistics - 9. ASIM-Fachtagung Simulation in Produktion und Logistik*. IPK, Berlin, 2000.
- [Sie02] SIEGRIST, K.: *Probabilistic/Statistic Object Library*. University of Alabama, Huntsville, <http://www.math.uah.edu/psol>, 2002.
- [Sto01] STORM, REGINA: *Wahrscheinlichkeitsrechnung, mathematische Statistik und statistische Qualitätskontrolle*. Fachbuchverlag, Leipzig, 2001.
- [VDI70] VDI-RICHTLINE 2411: *Begriffe und Erläuterungen im Förderwesen*. VDI-Verlag, Düsseldorf, 1970.
- [Vid91] VIDAKOVIC, B., MÜLLER, P.: *Wavelets for kids - a tutorial introduction*. Diskussionspapier, Institute of Statistics and Decision Sciences, Duke University, 1991.

- 
- [Völ02] VÖLKER, S.: *Reduktion von Simulationsmodellen zur simulationsbasierten Optimierung in der Termin- und Kapazitätsplanung*. Dissertation, TU Ilmenau, 2002.
- [Wan93] WANG, L., ALKON, D.L.: *Artificial neural networks : oscillations, chaos, and sequence processing*. IEEE Computer Society Press, Los Alamitos, Calif., 1993.
- [Web99] WEBER, M.: *Vorgabezeitermittlung mit künstlichen neuronalen Netzen für die variantenreiche Kleinserienmontage*. Jost Jetter Verlag, Heimsheim, 1999.
- [Wer74] WERBOS, P.: *Beyond regression: new tools for prediction and analysis in behavioral sciences*. Ph.D. thesis, Harvard University, 1974.
- [Wol93] WOLFRAM, S.: *Mathematica*. Addison-Wesley, Reading, 1993.
- [Zel94] ZELL, ANDREAS: *Simulation neuronaler Netze*. Addison-Wesley, Bonn, 1994.
- [Zel95] ZELL, A., MAMIER, G.: *SNNS User manual*. Universität Stuttgart, Stuttgart, Tübingen, 1995.

# Index

- Abgangsverteilung, 35
- Approximation
  - Eingangsdaten, 40
  - Zielgrößen, 40
- Attribut, 6
- Auslastung, 15
- Ausschleusung, 59
- Backpropagation, 24, 27
- Bedienstation, 68
- Blockierung, 17
- Diskrete Simulation, 18
- Drehtisch, 61
- Eingangsverteilung, 35
- Einschleusung, 51
- Gültigkeit, 13, 45
- Gammaverteilung, 42
- Gut, 1
- Gutstrom, 1
- Hubtisch, 61
- Klassifikation
  - von Materialfluss-Elementen, 37
  - von Systemen, 7
- Konfidenzintervall, 45
- Kreuzung, 61
- Materialfluss-System, 1
- Merkmal, 6
- Metamodell, 19
- Modell, 8, 9
- Modellierung, Systematik der, 10
- Modellklasse, 37
- Neuronale Netze
  - Aufbau, 22
  - Dämpfungsfaktor, 27
  - Topologie, 44
  - Training, 25
- Objekt, 6
- Quantil, 14
  - verstetigtes, 15
- Regalbediengerät, 64
- Stichprobenumfang, 45
- Streckenlast, 15
- System, 6
- Systemzustand, 7
- Trainingsdaten
  - Generierung, 42
  - Umfang, 43
- Trainingszyklus, 27
- Transporteinheit, 1
- Transportmatrix, 16
- Transportsystem, 1
- Transportsystem-Element, 1
  - Ordnung, 16
  - Typ, 16
- Übertrainieren (*overfitting*), 43
- Validierung, 13, 48
- Verifikation, 13, 47
- Wartesystem, 17
- Wegzeitmatrix, 16
- Zusammenführung, 51
- Zweipunktverteilung, 48
- Zwischenzeiten, 35